

No. 2

Supreme Court, U.S.  
FILED

942003 JUN 7 - 1995

IN THE OFFICE OF THE CLERK  
**Supreme Court of the United States**

OCTOBER TERM, 1994

LOTUS DEVELOPMENT CORPORATION,

*Petitioner,*

—v.—

BORLAND INTERNATIONAL, INC.,

*Respondent.*

ON PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES  
COURT OF APPEALS FOR THE FIRST CIRCUIT

**APPENDIX TO PETITION FOR A WRIT OF CERTIORARI**

CHARLES FRIED  
1575 Massachusetts Avenue  
Cambridge, Massachusetts 02138  
(617) 495-4636

ARTHUR R. MILLER  
1545 Massachusetts Avenue  
Cambridge, Massachusetts 02138  
(617) 495-4111

HENRY B. GUTMAN  
*Counsel of Record*  
KERRY L. KONRAD  
KAREN F. CONWAY  
JEFFREY E. OSTROW  
ANNE W. SALISBURY  
BAKER & BOTTS, L.L.P.  
885 Third Avenue  
New York, New York 10022  
(212) 705-5000

THOMAS M. LEMBERG  
LOTUS DEVELOPMENT  
CORPORATION  
55 Cambridge Parkway  
Cambridge, Massachusetts 02142  
(617) 693-8106

*Counsel for Petitioner*

300 PP

## INDEX TO APPENDIX

	PAGE
Opinion of the United States Court of Appeals for the First Circuit in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated March 9, 1995.	1a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated August 12, 1993, as amended August 19, 1993 ...	29a
Permanent Injunction entered by United States District Court of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated August 19, 1993 .....	69a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated June 30, 1993, as amended August 19, 1993 .....	71a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated July 31, 1992 .....	106a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated March 20, 1992 .....	145a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Paperback Software International</i> , dated June 28, 1990 .....	183a

	PAGE
The 1976 Copyright Act, 17 U.S.C. §§ 101, 102(a), 102(b), 103, 117 .....	270a
Affidavit of Mitchell D. Kapor, dated November 27, 1989, submitted in <i>Lotus Development Corporation</i> <i>v. Borland International, Inc. and Lotus</i> <i>Development Corporation v. Paperback Software</i> <i>International</i> .....	281a

UNITED STATES COURT OF APPEALS  
FOR THE FIRST CIRCUIT

---

No. 93-2214

---

LOTUS DEVELOPMENT CORPORATION,

*Plaintiff, Appellee.*

—v.—

BORLAND INTERNATIONAL, INC.,

*Defendant, Appellant.*

---

Appeal from the United States District Court  
for the District of Massachusetts  
[Hon. Robert E. Keeton *U.S. District Judge*]

---

Before  
TORRUELLA, *Chief Judge*,  
BOUDIN and STAHL, *Circuit Judges*.

---

*Gary L. Reback, with whom Peter N. Detkin, Michael Barclay, Isabella E. Fu, Wilson Sonsini Goodrich & Rosati, Peter E. Gelhaar, Katherine L. Parks, and Donnelly Conroy & Gelhaar, were on brief for appellant.*

*Matthew P. Poppel, et al., were on brief for Computer Scientists, amicus curiae.*

*Dennis S. Karjala*, and *Peter S. Menell*, on brief, amici curiae.

*Jeffrey C. Cannon* and *Baker Keaton Seibel & Cannon*, were on brief for Computer Software Industry Ass'n, amicus curiae.

*Laureen E. McGurk*, *David A. Rabin*, *Bryan G. Harrison* and *Morris Manning & Martin*, were on brief for Chicago Computer Soc., Diablo Users Group, Danbury Area Computer Soc., IBM AB Users Group, Kentucky-Indiana Personal Computer Users Group, Long Island PC Users Group, Napa Valley PC Users Group, Pacific Northwest PC Users Group, Palmetto Personal Computer Club, Philadelphia Area Computer Soc., Inc., Phoenix IBM PC Users Group, Pinellas IBM PC Users Group, Quad Cities Computer Soc., Quattro Pro Users Group, Sacramento PC Users Group, San Francisco PC Users Group, Santa Barbara PC Users Group, Twin Cities PC Users Group, and Warner Robbins Personal Computer Ass'n, amici curiae.

*Diane Marie O'Malley* and *Hanson Bridgett Marcus Vlahos & Rudy*, were on brief for Software Entrepreneurs' Forum, amicus curiae.

*Peter M.C. Choy*, was on brief for American Committee for Interoperable Systems, amicus curiae.

*Howard B. Abrams*, *Howard C. Anawalt*, *Stephen R. Barnett*, *Ralph S. Brown*, *Stephen L. Carter*, *Amy B. Cohen*, *Paul J. Heald*, *Peter A. Jaszi*, *John A. Kidwell*, *Edmund W. Kitch*, *Roberta R. Kwall*, *David L. Lange*, *Marshall Leaffer*, *Jessica D. Litman*, *Charles R. McManis*, *L. Ray Patterson*, *Jerome H. Reichman*, *David A. Rice*, *Pamela Samuelson*, *David J. Seipp*, *David E. Shipley*, *Lionel S. Sobel*, *Alfred C. Yen*, and *Diane L. Zimmerman*, were on brief for Copyright Law Professors, amicus curiae.

*Henry B. Gutman*, with whom *Kerry L. Konrad*, *Joshua H. Epstein*, *Kimberly A. Caldwell*, *O'Sullivan Graev & Karabell*, *Thomas M. Lemberg*, *James C. Burling*, and *Hale and Dorr*, were on brief for appellee.

*Morton David Goldberg*, *June M. Besek*, *David O. Carson*, *Jesse M. Feder*, *Schwab Goldberg Price & Dannay*, and

*Arthur R. Miller*, were on brief for Apple Computer, Inc., Digital Equip. Corp., International Business Machines Corp., and Xerox Corp., amici curiae.

*Jon A. Baumgarten*, *Proskauer Rose Goetz & Mendelsohn*, and *Robert A. Gorman*, were on brief for Adobe Systems, Inc., Apple Computer, Inc., Computer Associates Intern., Inc., Digital Equip. Corp., and International Business Machines Corp., amici curiae.

*Herbert F. Schwartz*, *Vincent N. Palladino*, *Susan Progoff*, *Fish & Neave*, *William J. Cheeseman*, and *Foley Hoag & Eliot*, were on brief for Computer and Business Equip. Mfrs. Ass'n, amicus curiae.

---

STAHL, Circuit Judge.

- This appeal requires us to decide whether a computer menu command hierarchy is copyrightable subject matter. In particular, we must decide whether, as the district court held, plaintiff-appellee Lotus Development Corporation's copyright in Lotus 1-2-3, a computer spreadsheet program, was infringed by defendant-appellant Borland International, Inc., when Borland copied the Lotus 1-2-3 menu command hierarchy into its Quattro and Quattro Pro computer spreadsheet programs. See *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F.Supp. 78 (D.Mass. 1992) ("Borland I"); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F.Supp. 203 (D.Mass. 1992) ("Borland II"); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F.Supp. 202 (D.Mass. 1993) ("Borland III"); *Lotus Dev. Corp. v. Borland Int'l Inc.*, 831 F.Supp. 223 (D.Mass. 1993) ("Borland IV").

### Background

Lotus 1-2-3 is a spreadsheet program that enables users to perform accounting functions electronically on a computer. Users manipulate and control the program via a series of menu commands, such as "Copy," "Print," and "Quit." Users choose commands either by highlighting them on the screen or by typing their first letter. In all, Lotus 1-2-3 has 469 commands arranged into more than 50 menus and submenus.

Lotus 1-2-3, like many computer programs, allows users to write what are called "macros." By writing a macro, a user can designate a series of command choices with a single macro keystroke. Then, to execute that series of commands in multiple parts of the spreadsheet, rather than typing the whole series each time, the user only needs to type the single pre-programmed macro keystroke, causing the program to recall and perform the designated series of commands automatically. Thus, Lotus 1-2-3 macros shorten the time needed to set up and operate the program.

Borland released its first Quattro program to the public in 1987, after Borland's engineers had labored over its development for nearly three-years. Borland's objective was to develop a spreadsheet program far superior to existing programs, including Lotus 1-2-3. In Borland's words, "[f]rom the time of its initial release . . . Quattro included enormous innovations over competing spreadsheet products."

The district court found, and Borland does not now contest, that Borland included in its Quattro and Quattro Pro version 1.0 programs "a *virtually identical* copy of the entire 1-2-3 menu tree." *Borland III*, 831 F.Supp. at 212 (emphasis in original). In so doing, Borland did not copy any of Lotus's underlying computer code; it copied only the words and structure of Lotus's menu command hierarchy. Borland included the Lotus menu command hierarchy in its programs to make them compatible with Lotus 1-2-3 so that spreadsheet users who were already familiar with Lotus 1-2-3 would be able to

switch to the Borland programs without having to learn new commands or rewrite their Lotus macros.

In its Quattro and Quattro Pro version 1.0 programs, Borland achieved compatibility with Lotus 1-2-3 by offering its users an alternate user interface, the "Lotus Emulation Interface." By activating the Emulation Interface, Borland users would see the Lotus menu commands on their screens and could interact with Quattro or Quattro Pro as if using Lotus 1-2-3, albeit with a slightly different looking screen and with many Borland options not available on Lotus 1-2-3. In effect, Borland allowed users to choose how they wanted to communicate with Borland's spreadsheet programs: either by using menu commands designed by Borland, or by using the commands and command structure used in Lotus 1-2-3 augmented by Borland-added commands.

Lotus filed this action against Borland in the District of Massachusetts on July 2, 1990, four days after a district court held that the Lotus 1-2-3 "menu structure, taken as a whole—including the choice of command terms [and] the structure and order of those terms," was protected expression covered by Lotus's copyrights. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37, 68, 70 (D.Mass.1990) ("*Paperback*").<sup>1</sup> Three days earlier, on the morning after the *Paperback* decision, Borland had filed a declaratory judgment action against Lotus in the Northern District of California, seeking a declaration of non-infringement. On September 10, 1990, the district court in California dismissed Borland's declaratory judgment action in favor of this action.

Lotus and Borland filed cross motions for summary judgment; the district court denied both motions on March 20, 1992, concluding that "neither party's motion is supported by the record." *Borland I*, 788 F.Supp. at 80. The district court invited the parties to file renewed summary judgment motions that would "focus their arguments more precisely" in light of rulings it had made in conjunction with its denial of their summary judgment motions. *Id.* at 82. Both parties filed

<sup>1</sup> Judge Keeton presided over both the *Paperback* litigation and this case.

renewed motions for summary judgment on April 24, 1992. In its motion, Borland contended that the Lotus 1-2-3 menus were not copyrightable as a matter of law and that no reasonable trier of fact could find that the similarity between its products and Lotus 1-2-3 was sufficient to sustain a determination of infringement. Lotus contended in its motion that Borland had copied Lotus 1-2-3's entire user interface and had thereby infringed Lotus's copyrights.

On July 31, 1992, the district court denied Borland's motion and granted Lotus's motion in part. The district court ruled that the Lotus menu command hierarchy was copyrightable expression because

[a] very satisfactory spreadsheet menu tree can be constructed using different commands and a different command structure from those of Lotus 1-2-3. In fact, Borland has constructed just such an alternate tree for use in Quattro Pro's native mode. Even if one holds the arrangement of menu commands constant, it is possible to generate literally millions of satisfactory menu trees by varying the menu commands employed.

*Borland II*, 799 F.Supp. at 217. The district court demonstrated this by offering alternate command words for the ten commands that appear in Lotus's main menu. *Id.* For example, the district court stated that "[t]he 'Quit' command could be named 'Exit' without any other modifications," and that "[t]he 'Copy' command could be called 'Clone,' 'Ditto,' 'Duplicate,' 'Imitate,' 'Mimic,' 'Replicate,' and 'Reproduce,' among others." *Id.* Because so many variations were possible, the district court concluded that the Lotus developers' choice and arrangement of command terms, reflected in the Lotus menu command hierarchy, constituted copyrightable expression.

In granting partial summary judgment to Lotus, the district court held that Borland had infringed Lotus's copyright in Lotus 1-2-3:

[A]s a matter of law, Borland's Quattro products infringe the Lotus 1-2-3 copyright because of (1) the extent of

copying of the "menu commands" and "menu structure" that is not *genuinely* disputed in this case, (2) the extent to which the copied elements of the "menu commands" and "menu structure" contain expressive aspects separable from the functions of the "menu commands" and "menu structure," and (3) the scope of those copied expressive aspects as an integral part of Lotus 1-2-3.

*Borland II*, 799 F.Supp. at 223 (emphasis in original). The court nevertheless concluded that while the Quattro and Quattro Pro programs infringed Lotus's copyright, Borland had not copied the entire Lotus 1-2-3 user interface, as Lotus had contended. Accordingly, the court concluded that a jury trial was necessary to determine the scope of Borland's infringement, including whether Borland copied the long prompts<sup>2</sup> of Lotus 1-2-3, whether the long prompts contained expressive elements, and to what extent, in any, functional constraints limited the number of possible ways that the Lotus menu command hierarchy could have been arranged at the time of its creation. *See Borland III*, 831 F.Supp. at 207. Additionally, the district court granted Lotus summary judgment on Borland's affirmative defense of waiver, but not on its affirmative defenses of laches and estoppel. *Borland II*, 799 F.Supp. at 222-23.

<sup>2</sup> Lotus 1-2-3 utilizes a two-line menu; the top line lists the commands from which the user may choose, and the bottom line displays what Lotus calls its "long prompts." The long prompts explain, as a sort of "help text," what the highlighted menu command will do if entered. For example, the long prompt for the "Worksheet" command displays the submenu that the "Worksheet" command calls up; it reads "Global, Insert, Delete, Column, Erase, Titles, Window, Status, Page." The long prompt for the "Copy" command explains what function the "Copy" command will perform: "Copy a cell or range of cells." The long prompt for the "Quit" command reads, "End 1-2-3 session (Have you saved your work?)."

Prior to trial, the parties agreed to exclude the copying of the long prompts from the case; Lotus agreed not to contend that Borland had copied the long prompts, Borland agreed not to argue that it had not copied the long prompts, and both sides agreed not to argue that the issue of whether Borland had copied the long prompts was material to any other issue in the case. *See Borland III*, 831 F.Supp. at 208.

Immediately following the district court's summary judgment decision, Borland removed the Lotus Emulation Interface from its products. Thereafter, Borland's spreadsheet programs no longer displayed the Lotus 1-2-3 menus to Borland users, and as a result Borland users could no longer communicate with Borland's programs as if they were using a more sophisticated version of Lotus 1-2-3. Nonetheless, Borland's programs continued to be partially compatible with Lotus 1-2-3, for Borland retained what it called the "Key Reader" in its Quattro Pro programs. Once turned on, the Key Reader allowed Borland's programs to understand and perform some Lotus 1-2-3 macros.<sup>3</sup> With the Key Reader on, the Borland programs used Quattro Pro menus for display, interaction, and macro execution, except when they encountered a slash ("/") key in a macro (the starting key for any Lotus 1-2-3 macro), in which case they interpreted the macro as having been written for Lotus 1-2-3. Accordingly, people who wrote or purchased macros to shorten the time needed to perform an operation in Lotus 1-2-3, could still use those macros in Borland's programs.<sup>4</sup> The district court permitted Lotus to file a supplemental complaint alleging that the Key Reader infringed its copyright.

The parties agreed to try the remaining liability issues without a jury. The district court held two trials, the Phase I trial covering all remaining issues raised in the original complaint (relating to the Emulation Interface) and the Phase II trial covering all issues raised in the supplemental complaint (relating to the Key Reader). At the Phase I trial, there were no live witnesses, although considerable testimony was presented in the form of affidavits and deposition excerpts. The district court ruled upon evidentiary objections counsel interposed. At the Phase II trial, there were two live witnesses,

<sup>3</sup> Because Borland's programs could no longer display the Lotus menu command hierarchy to users, the Key Reader did not allow debugging or modification of macros, nor did it permit the execution of most interactive macros.

<sup>4</sup> See *Borland IV*, 831 F.Supp. at 226-27, for a more detailed explanation of macros and the Key Reader.

each of whom demonstrated the programs for the district court.

After the close of the Phase I trial, the district court permitted Borland to amend its answer to include the affirmative defense of "fair use." Because Borland had presented all of the evidence supporting its fair-use defense during the Phase I trial, but Lotus had not presented any evidence on fair use (as the defense had not been raised before the conclusion of the Phase I trial), the district court considered Lotus's motion for judgment on partial findings of fact. See Fed.R.Civ.P. 52(c). The district court held that Borland had failed to show that its use of the Lotus 1-2-3 menu command hierarchy in its Emulation Interface was a fair use. See *Borland III*, 831 F.Supp. at 208.

In its Phase I-trial decision, the district court found that "each of the Borland emulation interfaces contains a virtually identical copy of the 1-2-3 menu tree and that the 1-2-3 menu tree is capable of a wide variety of expression." *Borland III*, 831 F.Supp. at 218. The district court also rejected Borland's affirmative defenses of laches and estoppel. *Id.* at 218-23.

In its Phase II-trial decision, the district court found that Borland's Key Reader file included "a virtually identical copy of the Lotus menu tree structure, but represented in a different form and with first letters of menu command names in place of the full menu command names." *Borland IV*, 831 F.Supp. at 228. In other words, Borland's programs no longer included the Lotus command terms, but only their first letters. The district court held that "the Lotus menu structure, organization, and first letters of the command names . . . constitute part of the protectable expression found in [Lotus 1-2-3]." *Id.* at 233. Accordingly, the district court held that with its Key Reader, Borland had infringed Lotus's copyright. *Id.* at 245. The district court also rejected Borland's affirmative defenses of waiver, laches, estoppel, and fair use. *Id.* at 235-45. The district court then entered a permanent injunction against Borland, *id.* at 245, from which Borland appeals.

This appeal concerns only Borland's copying of the Lotus menu command hierarchy into its Quattro programs and Borland's affirmative defenses to such copying. Lotus has not

cross-appealed; in other words, Lotus does not contend on appeal that the district court erred in finding that Borland had not copied other elements of Lotus 1-2-3, such as its screen displays.

## II.

### Discussion

On appeal, Borland does not dispute that it factually copied the words and arrangement of the Lotus menu command hierarchy. Rather, Borland argues that it "lawfully copied the unprotectable menus of Lotus 1-2-3." Borland contends that the Lotus menu command hierarchy is not copyrightable because it is a system, method of operation, process, or procedure foreclosed from protection by 17 U.S.C. § 102(b). Borland also raises a number of affirmative defenses.

#### A. Copyright Infringement Generally

To establish copyright infringement, a plaintiff must prove "(1) ownership of a valid copyright, and (2) copying of constituent elements of the work that are original." *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 361, 111 S.Ct. 1282, 1296, 113 L.Ed.2d 358 (1991); see also *Data Gen. Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1160 n. 19 (1st Cir.1994); *Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 605 (1st Cir.1988). To show ownership of a valid copyright and therefore satisfy *Feist's* first prong, a plaintiff must prove that the work as a whole is original and that the plaintiff complied with applicable statutory formalities. See *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1340 (5th Cir.1994). "In judicial proceedings, a certificate of copyright registration constitutes *prima facie* evidence of copyrightability and shifts the burden to the defendant to demonstrate why the copyright is not valid." *Bibbero Sys., Inc. v. Colwell Sys., Inc.*, 893 F.2d 1104, 1106 (9th Cir.1990); see also 17 U.S.C. § 410(c); *Folio Impressions, Inc. v. Byer California*, 937 F.2d 759, 763 (2d Cir.1991) (presumption of validity may be rebutted).

To show actionable copying and therefore satisfy *Feist's* second prong, a plaintiff must first prove that the alleged infringer copied plaintiff's copyrighted work as a factual matter; to do this, he or she may either present direct evidence of factual copying or, if that is unavailable, evidence that the alleged infringer had access to the copyrighted work and that the offending and copyrighted works are so similar that the court may infer that there was factual copying (i.e., probative similarity). *Engineering Dynamics*, 26 F.3d at 1340; see also *Concrete Mach.*, 843 F.2d at 606. The plaintiff must then prove that the copying of copyrighted material was so extensive that it rendered the offending and copyrighted works substantially similar. See *Engineering Dynamics*, 26 F.3d at 1341.

In this appeal, we are faced only with whether the Lotus menu command hierarchy is copyrightable subject matter in the first instance, for Borland concedes that Lotus has a valid copyright in Lotus 1-2-3 as a whole<sup>5</sup> and admits to factually copying the Lotus menu command hierarchy. As a result, this appeal is in a very different posture from most copyright-infringement cases, for copyright infringement generally turns on whether the defendant has copied protected expression as a factual matter. Because of this different posture, most copyright-infringement cases provide only limited help to us in deciding this appeal. This is true even with respect to those copyright-infringement cases that deal with computers and computer software.

<sup>5</sup> Computer programs receive copyright protection as "literary works." See 17 U.S.C. § 102(a)(1) (granting protection to "literary works") and 17 U.S.C. § 101 (defining "literary works" as "works . . . expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, phonorecords, film, tapes, disks, or cards, in which they are embodied" (emphasis added)); see also H.R.Rep. No. 1476, 94th Cong., 2d Sess. 54 (1976), reprinted in 1976 U.S.C.C.A.N. 5659, 5667 ("The term 'literary works' . . . includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.").

### B. Matter of First Impression

Whether a computer menu command hierarchy constitutes copyrightable subject matter is a matter of first impression in this court. While some other courts appear to have touched on it briefly in dicta, *see, e.g., Autoskill, Inc. v. National Educ. Support Sys., Inc.*, 994 F.2d 1476, 1495 n. 23 (10th Cir.), *cert. denied*, \_\_\_ U.S. \_\_\_, 114 S.Ct. 307, 126 L.Ed.2d 254 (1993), we know of no cases that deal with the copyrightability of a menu command hierarchy standing on its own (i.e., without other elements of the user interface, such as screen displays, in issue). Thus we are navigating in uncharted waters.

Borland vigorously argues, however, that the Supreme Court charted our course more than 100 years ago when it decided *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879). In *Baker v. Selden*, the Court held that Selden's copyright over the textbook in which he explained his new way to do accounting did not grant him a monopoly on the use of his accounting system.<sup>6</sup> Borland argues:

The facts of *Baker v. Selden*, and even the arguments advanced by the parties in that case, are identical to those in this case. The only difference is that the "user interface" of Selden's system was implemented by pen and paper rather than by computer.

To demonstrate that *Baker v. Selden* and this appeal both involve accounting systems, Borland even supplied this court with a video that, with special effects, shows Selden's paper forms "melting" into a computer screen and transforming into Lotus 1-2-3.

We do not think that *Baker v. Selden* is nearly as analogous to this appeal as Borland claims. Of course, Lotus 1-2-3 is a computer spreadsheet, and as such its grid of horizontal rows and vertical columns certainly resembles an accounting ledger or any other paper spreadsheet. Those grids, however, are not at issue in this appeal for, unlike Selden, Lotus does not claim to have a monopoly over its accounting system. Rather, this

<sup>6</sup> Selden's system of double-entry bookkeeping is the now almost-universal T-accounts system.

appeal involves Lotus's monopoly over the commands it uses to operate the computer. Accordingly, this appeal is not, as Borland contends, "identical" to *Baker v. Selden*.

### C. *Altai*

Before we analyze whether the Lotus menu command hierarchy is a system, method of operation, process, or procedure, we first consider the applicability of the test the Second Circuit set forth in *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir.1992).<sup>7</sup> The Second Circuit designed its *Altai* test to deal with the fact that computer programs, copyrighted as "literary works," can be infringed by what is known as "nonliteral" copying, which is copying that is paraphrased or loosely paraphrased rather than word for word. *See id.* at 701 (citing nonliteral-copying cases); *see also* 3 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* § 13.03[A][1] (1993). When faced with nonliteral-copying cases, courts must determine whether similarities are due merely to the fact that the two works share the same underlying idea or whether they instead indicate that the second author copied the first author's expression. The Second Circuit designed its *Altai* test to deal with this situation in the computer context, specifically with whether one computer program copied nonliteral expression from another program's code.

The *Altai* test involves three steps: abstraction, filtration, and comparison. The abstraction step requires courts to "dissect the allegedly copied program's structure and isolate each level of abstraction contained within it." *Altai* 982 F.2d at 707. This step enables courts to identify the appropriate framework within which to separate protectable expression from unprotected ideas. Second, courts apply a "filtration"

<sup>7</sup> We consider the *Altai* test because both parties and many of the *amici* focus on it so heavily. Borland, in particular, is highly critical of the district court for not employing the *Altai* test. Borland does not, however, indicate how using that test would have been dispositive in Borland's favor. Interestingly, Borland appears to contradict its own reasoning at times by criticizing the applicability of the *Altai* test.

step in which they examine "the structural components at each level of abstraction to determine whether their particular inclusion at that level was 'idea' or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain." *Id.* Finally, courts compare the protected elements of the infringed work (i.e., those that survived the filtration screening) to the corresponding elements of the allegedly infringing work to determine whether there was sufficient copying of protected material to constitute infringement. *Id.* at 710.

In the instant appeal, we are not confronted with alleged nonliteral copying of computer code. Rather, we are faced with Borland's deliberate, literal copying of the Lotus menu command hierarchy. Thus, we must determine not whether nonliteral copying occurred in some amorphous sense, but rather whether the literal copying of the Lotus menu command hierarchy constitutes copyright infringement.

While the *Altai* test may provide a useful framework for assessing the alleged nonliteral copying of computer code, we find it to be of little help in assessing whether the literal copying of a menu command hierarchy constitutes copyright infringement. In fact, we think that the *Altai* test in this context may actually be misleading because, in instructing courts to abstract the various levels, it seems to encourage them to find a base level that includes copyrightable subject matter that, if literally copied, would make the copier liable for copyright infringement.<sup>8</sup> While that base (or literal) level would not be at issue in a nonliteral-copying case like *Altai*, it is precisely what is at issue in this appeal. We think that abstracting menu command hierarchies down to their individual word and menu levels and then filtering idea from

<sup>8</sup> We recognize that *Altai* never states that every work contains a copyrightable "nugget" of protectable expression. Nonetheless, the implication is that for literal copying, "it is not necessary to determine the level of abstraction at which similarity ceases to consist of an 'expression of ideas,' because literal similarity by definition is always a similarity as to the expression of ideas." 3 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* § 13.03[A](2) (1993).

expression at that stage, as both the *Altai* and the district court tests require, obscures the more fundamental question of whether a menu command hierarchy can be copyrighted at all. The initial inquiry should not be whether individual components of a menu command hierarchy are expressive, but rather whether the menu command hierarchy as a whole can be copyrighted. *But see Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823 (10th Cir.1993) (endorsing *Altai*'s abstraction-filtration-comparison test as a way of determining whether "menus and sorting criteria" are copyrightable).

#### D. *The Lotus Menu Command Hierarchy: A "Method of Operation"*

Borland argues that the Lotus menu command hierarchy is uncopyrightable because it is a system, method of operation, process, or procedure foreclosed from copyright protection by 17 U.S.C. § 102(b). Section 102(b) states: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." Because we conclude that the Lotus menu command hierarchy is a method of operation, we do not consider whether it could also be a system, process, or procedure.

We think that "method of operation," as that term is used in § 102(b), refers to the means by which a person operates something, whether it be a car, a food processor, or a computer. Thus a text describing how to operate something would not extend copyright protection to the method of operation itself; other people would be free to employ that method and to describe it in their own words. Similarly, if a new method of operation is used rather than described, other people would still be free to employ or describe that method.

We hold that the Lotus menu command hierarchy is an uncopyrightable "method of operation." The Lotus menu command hierarchy provides the means by which users control and operate Lotus 1-2-3. If users wish to copy material, for example, they use the "Copy" command. If users wish to print material, they use the "Print" command. Users must use

the command terms to tell the computer what to do. Without the menu command hierarchy, users would not be able to access and control, or indeed make use of, Lotus 1-2-3's functional capabilities.

The Lotus menu command hierarchy does not merely explain and present Lotus 1-2-3's functional capabilities to the user; it also serves as the method by which the program is operated and controlled. The Lotus menu command hierarchy is different from the Lotus long prompts, for the long prompts are not necessary to the operation of the program; users could operate Lotus 1-2-3 even if there were no long prompts.<sup>9</sup> The Lotus menu command hierarchy is also different from the Lotus screen displays, for users need not "use" any expressive aspects of the screen displays in order to operate Lotus 1-2-3; because the way the screens look has little bearing on how users control the program, the screen displays are not part of Lotus 1-2-3's "method of operation."<sup>10</sup> The Lotus menu command hierarchy is also different from the underlying computer code, because while code is necessary for the program to work, its precise formulation is not. In other words, to offer the same capabilities as Lotus 1-2-3, Borland did not have to copy Lotus's underlying code (and indeed it did not); to allow users to operate its programs in substantially the same way, however, Borland had to copy the Lotus menu command hierarchy. Thus the Lotus 1-2-3 code is not a uncopyrightable "method of operation."<sup>11</sup>

<sup>9</sup> As the Lotus long prompts are not before us appeal, we take no position on their copyrightability, although we do note that a strong argument could be made that the brief explanations they provide "merge" with the underlying idea of explaining such functions. See *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir. 1967) (when the possible ways to express an idea are limited, the expression "merges" with the idea and is therefore uncopyrightable; when merger occurs, identical copying is permitted).

<sup>10</sup> As they are not before us on appeal, we take no position on whether the Lotus 1-2-3 screen displays constitute original expression capable of being copyrighted.

<sup>11</sup> Because the Lotus 1-2-3 code is not before us on appeal, we take no position on whether it is copyrightable. We note, however, that orig-

The district court held that the Lotus menu command hierarchy, with its specific choice and arrangement of command terms, constituted an "expression" of the "idea" of operating a computer program with commands arranged hierarchically into menus and submenus. *Borland II*, 799 F.Supp. at 216. Under the district court's reasoning, Lotus's decision to employ hierarchically arranged command terms to operate its program could not foreclose its competitors from also employing hierarchically arranged command terms to operate their programs, but it did foreclose them from employing the specific command terms and arrangement that Lotus had used. In effect, the district court limited Lotus 1-2-3's "method of operation" to an abstraction.

Accepting the district court's finding that the Lotus developers made some expressive choices in choosing and arranging the Lotus command terms, we nonetheless hold that that expression is not copyrightable because it is part of Lotus 1-2-3's "method of operation." We do not think that "methods of operation" are limited to abstractions; rather, they are the means by which a user operates something. If specific words are essential to operating something, then they are part of a "method of operation" and, as such, are unprotectable. This is so whether they must be highlighted, typed in, or even spoken, as computer programs no doubt will soon be controlled by spoken words.

The fact that Lotus developers could have designed the Lotus menu command hierarchy differently is immaterial to the question of whether it is a "method of operation." In other words, our initial inquiry is not whether the Lotus menu command hierarchy incorporates any expression.<sup>12</sup> Rather, our initial inquiry is whether the Lotus menu command hierarchy is a "method of operation." Concluding, as we do, that users operate Lotus 1-2-3 by using the Lotus menu command hier-

archical computer codes generally are protected by copyright. See, e.g., *Altai*, 982 F.2d at 702 ("It is now well settled that the literal elements of computer programs, i.e., their source and object codes, are the subject of copyright protection.") (citing cases).

<sup>12</sup> We think that the *Altai* test would contemplate this being the initial inquiry.

archy, and that the entire Lotus menu command hierarchy is essential to operating Lotus 1-2-3, we do not inquire further whether that method of operation could have been designed differently. The "expressive" choices of what to name the command terms and how to arrange them do not magically change the uncopyrightable menu command hierarchy into copyrightable subject matter.

Our holding that "methods of operation" are not limited to mere abstractions is bolstered by *Baker v. Selden*. In *Baker*, the Supreme Court explained that

the teachings of science and the rules and methods of useful art have their final end in application and use; and this application and use are what the public derive from the publication of a book which teaches them. . . . The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself. The object of the one is explanation; the object of the other is use. The former may be secured by copyright. The latter can only be secured, if it can be secured at all, by letters-patent.

*Baker v. Selden*, 101 U.S. at 104-05. Lotus wrote its menu command hierarchy so that people could learn it and use it. Accordingly, it falls squarely within the prohibition on copyright protection established in *Baker v. Selden* and codified by Congress in § 102(b).

In many ways, the Lotus menu command hierarchy is like the buttons used to control, say, a video cassette recorder ("VCR"). A VCR is a machine that enables one to watch and record video tapes. Users operate VCRs by pressing a series of buttons that are typically labelled "Record, Play, Reverse, Fast Forward, Pause, Stop/Eject." That the buttons are arranged and labeled does not make them a "literary work," nor does it make them an "expression" of the abstract "method of operating" a VCR via a set of labeled buttons. Instead, the buttons are themselves the "method of operating" the VCR.

When a Lotus 1-2-3 user chooses a command, either by highlighting it on the screen or by typing its first letter, he or

she effectively pushes a button. Highlighting the "Print" command on the screen, or typing the letter "P," is analogous to pressing a VCR button labeled "Play."

Just as one could not operate a buttonless VCR, it would be impossible to operate Lotus 1-2-3 without employing its menu command hierarchy. Thus the Lotus command terms are not equivalent to the labels on the VCR's buttons, but are instead equivalent to the buttons themselves. Unlike the labels on a VCR's buttons, which merely make operating a VCR easier by indicating the buttons' functions, the Lotus menu commands are essential to operating Lotus 1-2-3. Without the menu commands, there would be no way to "push" the Lotus buttons, as one could push unlabeled VCR buttons. While Lotus could probably have designed a user interface for which the command terms were mere labels, it did not do so here. Lotus 1-2-3 depends for its operation on use of the precise command terms that make up the Lotus menu command hierarchy.

One might argue that the buttons for operating a VCR are not analogous to the command for operating a computer program because VCRs are not copyrightable, where as computer programs are. VCRs may not be copyrighted because they do not fit within any of the § 102(a) categories of copyrightable works; the closest they come is "sculptural work." Sculptural works, however, are subject to a "useful-article" exception whereby "the design of a useful article . . . shall be considered a pictorial, graphic, or sculptural work only if, and only to the extent that, such design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article." 17 U.S.C. § 101. A "useful article" is "an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information." *Id.* Whatever expression there may be in the arrangement of the parts of a VCR is not capable of existing separately from the VCR itself, so an ordinary VCR would not be copyrightable.

Computer programs, unlike VCRs, are copyrightable as "literary works." 17 U.S.C. § 102(a). Accordingly, one might

argue, the "buttons" used to operate a computer program are not like the buttons used to operate a VCR, for they are not subject to a useful-article exception. The response, of course, is that the arrangement of buttons on a VCR would not be copyrightable even without a useful-article exception, because the buttons are an uncopyrightable "method of operation." Similarly, the "buttons" of a computer program are also an uncopyrightable "method of operation."

That the Lotus menu command hierarchy is a "method of operation" becomes clearer when one considers program compatibility. Under Lotus's theory, if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used. For example, if the user wanted the computer to print material, then the user would have to learn not just one method of operating the computer such that it prints, but many different methods. We find this absurd. The fact that there may be many different ways to operate a computer program, or even many different ways to operate a computer program using a set of hierarchically arranged command terms, does not make the actual method of operation chosen copyrightable; it still functions as a method for operating the computer and as such is uncopyrightable.

Consider also that users employ the Lotus menu command hierarchy in writing macros. Under the district court's holding, if the user wrote a macro to shorten the time needed to perform a certain operation in Lotus 1-2-3, the user would be unable to use that macro to shorten the time needed to perform that same operation in another program. Rather, the user would have to rewrite his or her macro using that other program's menu command hierarchy. This is despite the fact that the macro is clearly the user's own work product. We think that forcing the user to cause the computer to perform the same operation in a different way ignores Congress's direction in § 102(b) that "methods of operation" are not copyrightable. That programs can offer users the ability to write macros in many different ways does not change the fact that, once written, the macro allows the user to perform an operation automatically. As the Lotus menu command hierarchy

serves as the basis for Lotus 1-2-3 macros, the Lotus menu command hierarchy is a "method of operation."

In holding that expression that is part of a "method of operation" cannot be copyrighted, we do not understand ourselves to go against the Supreme Court's holding in *Feist*. In *Feist*, the Court explained:

The primary objective of copyright is not to reward the labor of authors, but to promote the Progress of Science and useful Arts. To this end, copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas and information conveyed by a work.

*Feist*, 499 U.S. at 349-50, 111 S.Ct. at 1290 (quotations and citations omitted). We do not think that the Court's statement that "copyright assures authors the right to their original expression" indicates that all expression is necessarily copyrightable; while original expression is necessary for copyright protection, we do not think that it is alone sufficient. Courts must still inquire whether original expression falls within one of the categories foreclosed from copyright protection by § 102(b), such as being a "method of operation."

We also note that in most contexts, there is no need to "build" upon other people's expression, for the ideas conveyed by that expression can be conveyed by someone else without copying the first author's expression.<sup>13</sup> In the context of methods of operation, however, "building" requires the use of the precise method of operation already employed; otherwise, "building" would require dismantling, too. Original developers are not the only people entitled to build on the methods of operation they create; anyone can. Thus, Borland may build on the method of operation that Lotus designed and may use the Lotus menu command hierarchy in doing so.

Our holding that methods of operation are not limited to abstractions goes against *Autoskill*, 994 F.2d at 1495 n. 23, in which the Tenth Circuit rejected the defendant's argument that

<sup>13</sup> When there are a limited number of ways to express an idea, however, the expression "merges" with the idea and becomes uncopyrightable. *Morrissey*, 379 F.2d at 678-79.

the keying procedure used in a computer program was an uncopyrightable "procedure" or "method of operation" under § 102(b). The program at issue, which was designed to test and train students with reading deficiencies, *id.* at 1481, required students to select responses to the program's queries "by pressing the 1, 2, or 3 keys." *Id.* at 1495 n. 23. The Tenth Circuit held that, "for purposes of the preliminary injunction, . . . the record showed that [this] keying procedure reflected at least a minimal degree of creativity," as required by *Feist* for copyright protection. *Id.* As an initial matter, we question whether a programmer's decision to have users select a response by pressing the 1, 2, or 3 keys is original. More importantly, however, we fail to see how "a student select[ing] a response by pressing the 1, 2, or 3 keys," *id.*, can be anything but an unprotectable method of operation.<sup>14</sup>

### III.

#### Conclusion

Because we hold that the Lotus menu command hierarchy is uncopyrightable subject matter, we further hold that Borland did not infringe Lotus's copyright by copying it. Accordingly, we need not consider any of Borland's affirmative defenses. The judgment of the district court is

*Reversed.*

*Concurrence follows.*

<sup>14</sup> The Ninth Circuit has also indicated in dicta that "menus, and keystrokes" may be copyrightable. *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1477 (9th Cir.), *cert. denied*, *BB Asset Management, Inc. v. Symantec Corp.*, \_\_\_ U.S. \_\_\_, 113 S.Ct. 198, 121 L.Ed.2d 141 (1992). In that case, however, the plaintiff did not show that the defendant had copied the plaintiff's menus or keystrokes, so the court was not directly faced with whether the menus or keystrokes constituted an unprotectable method of operation. *Id.*

BOUDIN, *Circuit Judge*, concurring.

The importance of this case, and a slightly different emphasis in my view of the underlying problem, prompt me to add a few words to the majority's tightly focused discussion.

### I.

Most of the law of copyright and the "tools" of analysis have developed in the context of literary works such as novels, plays, and films. In this milieu, the principal problem—simply stated, if difficult to resolve—is to stimulate creative expression without unduly limiting access by others to the broader themes and concepts deployed by the author. The middle of the spectrum presents close cases; but a "mistake" in providing too much protection involves a small cost: subsequent authors treating the same themes must take a few more steps away from the original expression.

The problem presented by computer programs is fundamentally different in one respect. The computer program is a *means* for causing something to happen; it has a mechanical utility, an instrumental role, in accomplishing the world's work. Granting protection, in other words, can have some of the consequences of *patent* protection in limiting other people's ability to perform a task in the most efficient manner. Utility does not bar copyright (dictionaries may be copyrighted), but it alters the calculus.

Of course, the argument *for* protection is undiminished, perhaps even enhanced, by utility: if we want more of an intellectual product, a temporary monopoly for the creator provides incentives for others to create other, different items in this class. But the "cost" side of the equation may be different where one places a very high value on public access to a useful innovation that may be the most efficient means of performing a given task. Thus, the argument for extending protection may be the same; but the stakes on the other side are much higher.

It is no accident that patent protection has preconditions that copyright protection does not—notably, the requirements

of novelty and non-obviousness—and that patents are granted for a shorter period than copyrights. This problem of utility has sometimes manifested itself in copyright cases, such as *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879), and been dealt with through various formulations that limit copyright or create limited rights to copy. But the case law and doctrine addressed to utility in copyright have been brief detours in the general march of copyright law.

Requests for the protection of computer menus present the concern with fencing off access to the commons in an acute form. A new menu may be a creative work, but over time its importance may come to reside more in the investment that has been made by *users* in learning the menu and in building their own mini-programs—macros—in reliance upon the menu. Better typewriter keyboard layouts may exist, but the familiar QWERTY keyboard dominates the market because that is what everyone has learned to use. See P. David, *CLIO and the Economics of QWERTY*, 75 *Am.Econ.Rev.* 332 (1985). The QWERTY keyboard is nothing other than a menu of letters.

Thus, to assume that computer programs are just one more new means of expression, like a filmed play, may be quite wrong. The “form”—the written source code or the menu structure depicted on the screen—look hauntingly like the familiar stuff of copyright; but the “substance” probably has more to do with problems presented in patent law or, as already noted, in those rare cases where copyright law has confronted industrially useful expressions. Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit.

All of this would make no difference if Congress had squarely confronted the issue, and given explicit directions as to what should be done. The Copyright Act of 1976 took a different course. While Congress said that computer programs might be subject to copyright protection, it said this in very general terms; and, especially in § 102(b), Congress adopted a string of exclusions that if taken literally might easily seem to exclude most computer programs from protection. The only

detailed prescriptions for computers involve narrow issues (like back-up copies) of no relevance here.

Of course, one could still read the statute as a congressional command that the familiar doctrines of copyright law be taken and applied to computer programs, in cookie cutter fashion, as if the programs were novels or play scripts. Some of the cases involving computer programs embody this approach. It seems to be mistaken on two different grounds: the tradition of copyright law, and the likely intent of Congress.

The broad-brush conception of copyright protection, the time limits, and the formalities have long been prescribed by statute. But the heart of copyright doctrine—what may be protected and with what limitations and exceptions—has been developed by the courts through experience with individual cases. B. Kaplan, *An Unhurried View of Copyright* 40 (1967). Occasionally Congress addresses a problem in detail. For the most part the interstitial development of copyright through the courts is our tradition.

Nothing in the language or legislative history of the 1976 Act, or at least nothing brought to our attention, suggests that Congress meant the courts to abandon this case-by-case approach. Indeed, by setting up § 102(b) as a counterpoint theme, Congress has arguably recognized the tension and left it for the courts to resolve through the development of case law. And case law development is *adaptive*: it allows new problems to be solved with help of earlier doctrine, but it does not preclude new doctrines to meet new situations.

## II.

In this case, the raw facts are mostly, if not entirely, undisputed. Although the inferences to be drawn may be more debatable, it is very hard to see that Borland has shown any interest in the Lotus menu except as a fall-back option for those users already committed to it by prior experience or in order to run their own macros using 1-2-3 commands. At least for the amateur, accessing the Lotus menu in the Borland Quattro or Quattro Pro program takes some effort.

Put differently, it is unlikely that users who value the Lotus menu for its own sake—independent of any investment they have made themselves in learning Lotus' commands or creating macros dependent upon them—would choose the Borland program in order to secure access to the Lotus menu. Borland's success is due primarily to other features. Its rationale for deploying the Lotus menu bears the ring of truth.

Now, any use of the Lotus menu by Borland is a commercial use and deprives Lotus of a portion of its "reward," in the sense that an infringement claim if allowed would increase Lotus' profits. But this is circular reasoning: broadly speaking, every limitation on copyright or privileged use diminishes the reward of the original creator. Yet not every writing is copyrightable or every use an infringement. The provision of reward is one concern of copyright law, but it is not the only one. If it were, copyrights would be perpetual and there would be no exceptions.

The present case is an unattractive one for copyright protection of the menu. The menu commands (*e.g.*, "print," "quit") are largely for standard procedures that Lotus did not invent and are common words that Lotus cannot monopolize. What is left is the particular combination and sub-grouping of commands in a pattern devised by Lotus. This arrangement may have a more appealing logic and ease of use than some other configurations; but there is a certain arbitrariness to many of the choices.

If Lotus is granted a monopoly on this pattern, users who have learned the command structure of Lotus 1-2-3 or devised their own macros are locked into Lotus, just as a typist who has learned the QWERTY keyboard would be the captive of anyone who had a monopoly on the production of such a keyboard. Apparently, for a period Lotus 1-2-3 has had such sway in the market that it has represented the *de facto* standard for electronic spreadsheet commands. So long as Lotus is the superior spreadsheet—either in quality or in price—there may be nothing wrong with this advantage.

But if a better spreadsheet comes along, it is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an

investment in learning made by the users and not by Lotus. Lotus has already reaped a substantial reward for being first; assuming that the Borland program is now better, good reasons exist for freeing it to attract old Lotus customers: to enable the old customers to take advantage of a new advance, and to reward Borland in turn for making a better product. If Borland has not made a better product, then customers will remain with Lotus anyway.

Thus, for me the question is not whether Borland should prevail but on what basis. Various avenues might be traveled, but the main choices are between holding that the menu is not protectable by copyright and devising a new doctrine that Borland's use is privileged. No solution is perfect and no intermediate appellate court can make the final choice.

To call the menu a "method of operation" is, in the common use of those words, a defensible position. After all, the purpose of the menu is not to be admired as a work of literary or pictorial art. It is to transmit directions from the user to the computer, *i.e.*, to operate the computer. The menu is also a "method" in the dictionary sense because it is a "planned way of doing something," an "order or system," and (aptly here) an "orderly or systematic arrangement, sequence or the like." *Random House Webster's College Dictionary* 853 (1991).

A different approach would be to say that Borland's use is privileged because, in the context already described, it is not seeking to appropriate the advances made by Lotus' menu; rather, having provided an arguably more attractive menu of its own, Borland is merely trying to give former Lotus users an option to exploit their own prior investment in learning or in macros. The difference is that such a privileged use approach would not automatically protect Borland if it had simply copied the Lotus menu (using different codes), contributed nothing of its own, and resold Lotus under the Borland label.

The closest analogue in conventional copyright is the fair use doctrine. *E.g.*, *Harper & Row, Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 105 S.Ct. 2218, 85 L.Ed.2d 588 (1985). Although invoked by Borland, it has largely been brushed aside in this case because the Supreme Court has said

that it is "presumptively" unavailable where the use is a "commercial" one. *See id.* at 562, 105 S.Ct. at 2231-32. *But see Campbell v. Acuff-Rose Music, Inc.*, \_\_\_ U.S. \_\_\_, \_\_\_, 114 S.Ct. 1164, 1174, 127 L.Ed.2d 500 (1994). In my view, this is something less than a definitive answer; "presumptively" does not mean "always" and, in any event, the doctrine of fair use was created by the courts and can be adapted to new purposes.

But a privileged use doctrine would certainly involve problems of its own. It might more closely tailor the limits on copyright protection to the reasons for limiting that protection; but it would entail a host of administrative problems that would cause cost and delay, and would also reduce the ability of the industry to predict outcomes. Indeed, to the extent that Lotus' menu is an important standard in the industry, it might be argued that any use ought to be deemed privileged.

In sum, the majority's result persuades me and its formulation is as good, if not better, than any other that occurs to me now as within the reach of courts. Some solutions (*e.g.*, a very short copyright period for menus) are not options at all for courts but might be for Congress. In all events, the choices are important ones of policy, not linguistics, and they should be made with the underlying considerations in view.

UNITED STATES DISTRICT COURT  
D. MASSACHUSETTS

Civ. A. No. 90-11662-K

Aug. 12, 1993.

As Amended Aug. 19, 1993.

Permanent Injunction Aug. 19, 1993.

---

LOTUS DEVELOPMENT CORPORATION,

*Plaintiff.*

—v.—

BORLAND INTERNATIONAL, INC.,

*Defendant.*

---

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, MA, Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, MA, Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, CA, David L. Hayes, Mitchell Zimmerman, Fenwick & West, Palo Alto, CA, Peter E. Gelhaar, Donnelly, Conroy & Gelhaar, Boston, MA, Gary L. Reback, Nina F. Locker, Peter N. Detkin, David A. Priebe, James A. DiBoise, Wilson, Soncini, Goodrich & Rosati, Palo Alto, CA, Laurence H. Tribe, Cambridge, MA, *for defendant.*

---

KEETON, District Judge.

A nonjury trial was held in this action in two phases. Phase I was tried on February 1-3, 1993; Phase II, on March 31-April 2, 1993. An Opinion of June 30, 1993 (the "Phase I Opinion") stated the court's findings and conclusions on issues raised in Phase I of the trial. The present Opinion states my findings and conclusions for the Phase II trial.

## I. Introduction.

This Opinion assumes the reader's familiarity with, and follows the terminology set forth in, the Phase I Opinion. Background information appears also in *Lotus Dev. Corp. v. Borland Int'l Inc.*, 799 F.Supp. 203 (D.Mass.1992) ("*Borland II*"), *Lotus Dev. Corp. v. Borland Int'l Inc.*, 788 F.Supp. 78 (D.Mass.1992) ("*Borland I*"), and *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass.1990) ("*Paperback*").

As explained in the Phase I Opinion, the Phase I trial concerned issues of infringement raised in Lotus's original complaint, *ie.*, infringement by Borland's copying of Lotus's menu tree into the 1-2-3 emulation interface of Borland's Quattro and certain Quattro Pro spreadsheet programs.

Phase II of the trial concerns the full range of liability issues, including affirmative defenses, with respect to Lotus's supplemental complaint. The supplemental complaint alleges that Borland's "Key Reader" infringes Lotus's copyrights in its 1-2-3 programs.

## II. Does Key Reader Infringe?

### A. What is Key Reader?

The Phase I Opinion and the court's earlier Memorandum and Orders have focused on the 1-2-3 emulation interfaces in the Quattro and Quattro Pro programs. After executing the command that invokes the emulation interface in one of Borland's products, the program uses copies of the Lotus menu

tree for display to the user, interaction with the user (*i.e.*, interpreting the keystrokes typed by the user), and running macros.

The Key Reader is not a part of the emulation interface. The Key Reader feature may be turned on while the user continues to use another (*e.g.*, the Quattro) menu tree. When Key Reader has been turned on, the program continues to behave as it had before, with one exception. With Key Reader on, when the program encounters a slash key ("/") in a macro, the program interprets everything that follows the slash key as though it were part of a macro written for use with Lotus 1-2-3. Thus with Key Reader on, the program uses Quattro Pro menus for display, interaction and macro execution, except when a "/" key is encountered in a macro. Then, the program runs macros as though they were written using the 1-2-3 menu tree.

A more detailed explanation of 1-2-3 "macros" is essential to explaining Key Reader.

Generally, a 1-2-3 macro consists of some text contained in a spreadsheet cell. When the macro is invoked, the program begins at the left end of the text and treats each character in the text as though the user had typed that character into the 1-2-3 program. For example, consider a macro that consists of the text "/wp". When the macro is invoked, the program behaves precisely as though the user had typed into the regular interface a "/" (calling up the first menu) followed by "w" (selecting the "Worksheet" branch of the menu tree) followed by "p" (selecting the "Page" leaf; this executable operation inserts a page break into the spreadsheet).

In writing a macro, the user may use special commands not found in the 1-2-3 menu tree. For example, the character "" in a macro is interpreted as though the user struck the Enter key on the keyboard. Consider a macro consisting of the text "/rfc"". When invoked, this macro has the same effect as the user's typing into the 1-2-3 interface "/", "r", "f", "c", followed by striking the Enter key twice. That is, the program follows the "Range" branch, then the "Format" branch, then selects the "Currency" leaf. When the "Currency" leaf has

been selected, the program asks the user to specify the number of decimal places and the block of cells whose appearance is to be altered to that of monetary units. The first "" is equivalent to striking the Enter key; the program accepts the default number of decimal places (two). The program then asks for the range of cells. The second "" is equivalent to striking the Enter key a second time; the program accepts the default range (one cell).

Other sequences of symbols in a Lotus macro have a special meaning. Of particular interest is "{?}". When the program encounters this sequence of characters in a macro, the program pauses and waits for the user to strike keys appropriate for the menu tree. Thus, consider a Lotus macro consisting of the text "/rf{?}"". When invoked, this macro would follow the "r" command ("Range" from the first level of the Lotus menu tree), then the "f" command ("Format" in the Lotus menu tree). The program then encounters the "{?}" characters. Here, the program pauses and allows the user to input a menu command. The program will interpret what the user inputs as a choice from the Lotus 1-2-3 submenu corresponding to "Range" "Format". Thus, the user might strike the "c" key (followed by Enter) to format a cell as currency. If the user strikes the Escape key instead of striking "c", the program backs up the menu tree from the "Format" submenu to the "Range" submenu. The user may continue maneuvering up and down the Lotus 1-2-3 menu tree until striking the Enter key. When the user strikes the Enter key, the program returns to reading the text of the macro as though the user were continuing by striking the keys found in the remaining text of the macro. A macro consisting only of "/{?}" permits a user to select any executable operation by typing keys precisely as though the user were using one of Borland's emulation interfaces, but without display of the menus.

Lotus 1-2-3 macros may also contain more advanced commands. Thus, a macro may contain text such as "{if . . .}" or "{let . . .}". In these strings of characters, the ellipses refer to additional text or "arguments" the user would provide. When the program encounters an advanced command such as

this, the program follows whatever procedures that command calls for. For example, the text "{blank A1..G45}" would cause the program to erase each of the cells from "A1" (first column and first row) through "G45" (seventh column and forty-fifth row). "if", "let", and "blank" are all labels in the Lotus 1-2-3 macro language that do not appear in the 1-2-3 menu tree.

In sum, when Key Reader is on and a slash key is encountered in a macro, the program follows the text of the macro as though the characters were being typed during the program's use of a copy of the 1-2-3 menu tree (and with some other means for interpreting the various special Lotus 1-2-3 macro language commands). When Key Reader is off, or no "/" key is encountered in a macro, the program interprets macros by reference to a different (*e.g.*, the Quattro) menu tree. With Key Reader off, Borland's programs cannot correctly interpret 1-2-3 macros.

Quattro and Quattro Pro version 1.0 did not contain the Key Reader feature; they contained only the emulation interface. Thus in these programs, the user could execute Lotus macros only when the user was employing the emulation interface; *ie.*, using Lotus 1-2-3 menus for display, interpreting user commands, and executing macros.

The Key Reader was first introduced in Quattro Pro version 2.0. Quattro Pro version 2.0, 3.0, and 4.0 contain *both* the emulation interface and the Key Reader feature (which is used from the native menus).

Borland removed the emulation interface from Quattro Pro version 4.01 (and subsequent releases of Quattro Pro) after the *Borland II* decision in this case allowed partial summary judgment for Lotus. Thus, Quattro Pro versions 4.01, SE, and Quattro Pro for Windows contain the Key Reader feature but have no 1-2-3 emulation interface.

### **B. Copying.**

In developing Key Reader, Borland modified portions of the earlier Quattro Pro programs that contained the full 1-2-3

emulation interface. Accordingly, I begin with a review of how the Lotus menu tree was copied into the Quattro Pro emulation interface before analyzing how that was, in turn, incorporated into Key Reader.

### 1. The 123.MU file.

The actual menu tree for the Quattro Pro emulation interfaces (versions 1.0, 2.0, 3.0, and 4.0) was in a file labeled "123.MU". Thus, when the program was running and the user had selected the 1-2-3 emulation interface, the program would refer to the 123.MU file to determine the form of menu commands and menu structure used to present the set of executable operations to the user, interpret user commands, and interpret macros. If the user had selected a different interface, e.g., one using the Quattro native menus, the program would refer to a different file for the menu commands and menu structure for display, interpretation of commands, and macro execution.

A text print-out of portions of the 123.MU file is in evidence under seal as Exhibit 13. The parties have included, however, the first page of this print-out in the public record. See Docket 311, Exh. A. An examination of the print-out demonstrates that the entire Lotus menu tree is copied into the file, with differences in indentation detailing the menu structure. That is, the structure of the menus and submenus is recorded in the file by changes in indentation. As explained in the Opinion for Phase I, when the program is using the 123.MU file as the source for its menus, it presents to the user a virtually identical copy of the Lotus menu tree (with Borland's additional menu commands inserted into the tree).

### 2. Key Reader files.

As the above explanation of macros suggests, interpretation of Lotus macros requires that the program treat the characters in the macro text as though they were keystrokes into a spreadsheet program using the Lotus menu tree (except for the characters in the macro text that are special commands in the Lotus macro language).

I find that, to implement the Key Reader (or at least that portion of the Key Reader that interprets characters from the Lotus menu tree), Borland began with the 123.MU file from the emulation interface. Borland then prepared a new file by reproducing the old 123.MU file but with only the first letter of each menu command name where the entire Lotus menu command name appears in the old 123.MU file. See Warfield Dep'n. IX:43. Put another way, the point is that to implement Key Reader Borland used a program file containing the same copy of the 1-2-3 menu tree structure and commands that Borland had used in its emulation interface, but with each menu command name stripped of everything after the first letter. Borland then appended this copy of the "stripped menu tree" to its quattro.mu file.

At various stages, Borland introduced other changes in how the menu structure is recorded in the stripped file, altering parentheses or indentation, etc. For example, I find that Borland may have altered the symbols used to record Lotus's menu structure, inserted the word "PICKLETTER" before the first letter of the menu command, or made other changes. Compare Exhibit 13 with Exhibit 517; see Warfield Dep'n, IX:43 at 148-49. Some of these things may actually appear in the old 123.MU file—Exhibit 13 contains only portions of the old 123.MU file. See Warfield Dep'n, IX:43 at 107-109, 148-49. In any event, I have considered the file and possible changes. I find that none of these changes is material to the scope or nature of copying from the Lotus 1-2-3 program.

In short, I find that the Key Reader file contains a virtually identical copy of the Lotus menu tree structure, but represented in a different form and with first letters of menu command names in place of the full menu command names.

Borland contends that the command letters copied from Lotus 1-2-3 that are in Borland's Key Reader files are not in the same order as displayed on the screen in Lotus 1-2-3. That is, if one reads sequentially down the Key Reader file, one encounters the Lotus commands in a different order than when reading the menus from the display screen while running Lotus 1-2-3. This is true, however, only in the sense that

the menu structure of Lotus 1-2-3 is represented in a different way in the Key Reader *file* than on the Lotus 1-2-3 *display screens*; in the file, the structure is detailed by differences in indentation (or other means) rather than through display on the screen. I find that the file (in evidence under seal, Exhibit 517) fully delineates a virtually identical copy of the menu structure of Lotus 1-2-3 including the first letter of each menu command in the corresponding location in the copy of the menu structure.

Because these menus are used in interpreting macros but are never fully displayed to the user, the parties have sometimes referred to these as "phantom menus."

In sum, to interpret macros, Borland's programs use a file with phantom menus consisting of a virtually identical copy of the Lotus menu tree that Borland used for its emulation interface, but with only the first letter of each menu command name where the complete menu command name previously appeared.

### C. Copyrightability Issues.

#### 1. Renewed and new arguments concerning definition of the idea.

Borland contends that copyright protection covering the structure of the 1-2-3 menu tree and the first letters of the commands in the 1-2-3 menu tree (*ie.*, Borland's phantom menus) would be equivalent to copyright protection for a "system" or "method" of communication between the user and the program. Thus, Borland argues that copying of the 1-2-3 menu tree structure and first letters of command names is a necessary part of any *system* for interpreting Lotus 1-2-3 macros. Citing the proposition that copyright law does not protect a "system", *see* 17 U.S.C. § 102(b), Borland argues that the Lotus copyright cannot extend to Borland's phantom menus.

Lotus responds that Borland has not proved that copying of any part of the menu tree is necessary for running or translating macros.

As a preliminary matter, I observe that the parties, witnesses (and the court, in some instances) have not used the terms "macro translation," "macro conversion," "macro execution," and "macro compatibility" with precision during the Phase II trial and earlier proceedings. In order to avoid possible future misunderstanding, I will explain the ambiguity in use of these terms before proceeding farther.

Macro "conversion" refers to translating a macro that was written using Lotus 1-2-3 into a macro written in a form for use in another program. For example, the macro "/rfe~" in Lotus may be translated, using the macro translation assistant in Excel 2.1, into an Excel 2.1 macro having the text:

```
= SET.NAME("Selection—Save", SELECTION( ))
= SET.NAME("Range3", SELECTION( ))
= DISPLAY(FALSE)
= SELECT(Range3)
= FORMAT.NUMBER
("$#,##0.00;($#,##0.00)")
= RETURN( )
```

This macro text is written in the Excel macro language and is executed by reference to the Excel menus. Thus, formatting a single cell to be displayed in monetary units with two decimal places (the result of the "/rfe~" macro in Lotus) may be achieved in Excel 2.1 by selecting a single cell in an Excel spreadsheet followed by selection of the "Format" menu command, the "Number" command in the submenu of Format, then the "\$#,##0.00;(\$#,##0.00)" (currency) leaf (compare with the second to last line of the Excel macro).

This facility for conversion of macros may also be referred to as a "one-time macro translator." The translator takes a Lotus macro and converts it into a macro written in a different macro language, for use with a different menu tree. Once the translation has been made, a user may run the translated macro as frequently as desired in the other program. Because the macro is written in a different macro language, the program need not refer to a copy of Lotus's menu tree to run the (translated) macro. Thus in *Borland II*, I observed that:

I need not and do not decide whether Borland is prohibited from reading and interpreting macros that have been created by users of 1-2-3. Had Borland created a program that read users' 1-2-3 macros and converted them to macros for use in the Quattro programs' native modes, so that they could be interpreted, executed, modified, debugged, etc. by resort to Borland's command hierarchy, that would have presented a different case from the one now before me.

799 F.Supp. at 214. In other words, I did not decide issues concerning one-time translation of macros into a different macro language, such that the macro could be executed, modified, and debugged without reference to copies of the Lotus menu tree.

Alternatively, consider a continuous macro "interpreter." This facility executes Lotus macros by referring to copies of the Lotus menu tree contained within the program. Macro interpretation may thus be viewed as continuous "translation" of macros as each macro is executed, or "on-the-fly" interpretation. Each time a macro is executed, modified, or debugged, the macro remains written in the Lotus macro language. Each time a macro is executed, modified or debugged, the program refers to the Lotus menu tree or structure.

Both one-time conversion and on-the-fly interpretation may be referred to as "macro translation," as providing an ability to "run" Lotus macros, and as providing "macro compatibility" with Lotus. For both, the user may begin with a Lotus macro and run the macro either by translating it first (one-time translation), or by parsing it with reference to the Lotus menu tree structure each time the user runs the macro (on-the-fly).

An example of the confusion in terms is demonstrated by testimony from Borland's Chief Executive Officer, Phillip Kahn. Kahn testified that "macro translation was not a viable way to do things." I:13 at 100. From the context of this testimony, I understand the testimony to refer to one-time "macro translation" rather than what Borland now refers to as (on-the-fly) "macro translation" performed by Key Reader.

Borland's Key Reader does not perform one-time translation such as Excel 2.1 does. Rather, Key Reader interprets macros on-the-fly, by reference to "phantom" menus that contain a copy of the Lotus menu tree.

In clarifying this terminology, I do not decide whether copying of the Lotus menu structure for the purpose of one-time translation rather than on-the-fly interpretation should be accorded different treatment under copyright law. Also, I do not decide any issue concerning who (if anyone) owns the copyright in a macro written by a user in reliance on expression found in the Lotus 1-2-3 menu tree, or who (if anyone) owns the copyright in a translation into another language of a macro written in reliance on Lotus 1-2-3. These issues are not before the court. Rather, I make clear that I decide *only* those issues raised by Borland's Key Reader; *ie.*, only issues involved in on-the-fly interpretation using Borland's "phantom" menus.

Having clarified the terminology, I now reject Lotus's argument that on-the-fly macro interpretation does not require copying from the Lotus menu structure and first letters of the command names. To interpret a macro, the program must use the Lotus 1-2-3 menu structure. If a program did not have a representation of the 1-2-3 menu hierarchy somewhere within the program code (or in a file that is used by the code), then there is no way that the program could understand that "rfc" refers to a path through a menu tree to the specific executable operation that changes a cell or cells appearance to monetary units (*i.e.*, a path through the range and format menus to the currency leaf). Whether the menu tree is copied into a file or directly into the code of a program does not make a difference. The scope of copying of the menu tree is the same whether done in one computer language, a different language, or in a file. Accordingly, I reject Lotus's contention that no part of the 1-2-3 menu tree need be copied to interpret 1-2-3 macros.

Nevertheless, I also reject Borland's contention that the menu tree structure and first letters of the menu commands constitute "system" or "method," as those terms are used in copyright law.

Of course, it is possible to think of the Lotus 1-2-3 menu tree as a "system" for translating user keystrokes into executable operations (whether the keystrokes are typed into the keyboard or are taken from the text of a macro command). In the same way, a book may be thought of as a system for communicating various ideas and images to readers. Indeed, the only way of conveying each of the exact images of "Gone With the Wind" in the same order as in "Gone With the Wind" is to copy at least a great deal of the nonliteral expression in the book, if not the precise words of the book as well. Nevertheless, copyright protection extends both to the specific words of "Gone With the Wind" and to nonliteral elements of the novel. Accordingly, the ability to describe a work as a "system" is not decisive of whether the work is a "system," or instead is protected expression, under copyright law.

In calling the copying that is contained in the phantom menu tree a "system," one may attempt to distinguish the copying found in the phantom menus from copying of a book on the following grounds. A book communicates to the reader. The phantom menus, however, permit the user to communicate commands to the program. Thus, one may contend that the phantom menu trees constitute a method or system for the user to communicate executable commands rather than a system (such as a book) for communicating thoughts to the reader.

For the following reasons, I reject this basis for concluding that, in creating the phantom menus, Borland copied a system and not its expression. First, like Lotus's menu tree, the protected expression of a compilation may be viewed as a system for accessing information. The purpose of a compilation is to communicate facts. The specific facts communicated, however, are not copyrightable. Copyright protects only the selection, arrangement and manner of presentation of the facts (to the extent that those elements meet the other requirements for copyrightability). See *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991). The selection, arrangement, and manner of presentation in a compilation may provide the user with a *method* or

systematic manner of accessing the (uncopyrightable) facts. Thus, copyright law protects only that part of a compilation that the reader actually *uses* for selection of facts that the reader wants to know. Nevertheless, the expressive aspects of a compilation remain copyrightable.

Accordingly, I conclude that the fact that the copied menu structure and first letters of the menu commands may be used to specify executable operations does not bar a finding that these elements are copyrightable. For the reasons explained in detail in Sections II.A.1 and II.B2.b of the Phase I Opinion, the structure of the menu tree including its designated keys for invoking commands (*i.e.* what Borland copied into the phantom menus) may also be viewed, in a light favorable to Borland, as a type of selection and arrangement of the executable operations in Lotus 1-2-3. The fact that the phantom menus may be used by a "reader" (whether directly or through the text of a macro) to access Lotus 1-2-3's executable operations does not compel a conclusion that this constitutes a system. See also *Paperback*, 740 F.Supp. at 72-73 (rejecting argument that the macro language is a non-copyrightable element of the program).

Similarly, a program's computer code may be described as a system. Computer code consists of a series of commands that the user issues to the computer by running the program. The commands are written in such a way that the computer code as a whole is a system for communicating the program's functionality to the computer. Nevertheless, Borland cannot and does not dispute that computer code is copyrightable.

Borland's contention that the menu tree structure is a system, or is strictly utilitarian in nature, raises again the same problem of defining the "idea," "system," "process," "procedure," or "method" of Lotus 1-2-3 that I have addressed since the beginning of this case. Borland wishes the court to define the "idea," "system," "process," "procedure," or "method" of Lotus 1-2-3 as including the ability to interpret macros written for use with Lotus 1-2-3. I rejected this contention in deciding the parties' cross motions for summary judgment, see 799 F.Supp. at 212-14, 216-17, and I reject this

argument for the same reasons here. The fact that users of Lotus 1-2-3 have created macros in reliance on expressive aspects of Lotus 1-2-3 does not convert that expression into a part of the "system." That Borland wishes to copy protected expression contained in Lotus's menu tree for what Borland contends is a utilitarian purpose also does not turn that expression into a "system" under copyright law.

*See Id* at 213-14. *See also Mazer v. Stein*, 347 U.S. 201 [74 S.Ct. 460, 98 L.Ed. 630] (1954) (use of statue as lamp does not bar copyright protection on expression in statue); 1 Nimmer on Copyright § 2.18[D] at 2-207 ("If, however, there is a copying of the copyrightable expression, then an infringement should be found, even if the defendant employs the material for use rather than explanation.").

Borland nevertheless contends that according copyright protection to the menu structure and first letter of the command names would impermissibly protect a "discretionary pattern of events or processes." Section 102(b) provides:

In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

17 U.S.C. § 102(b). Borland quotes a proposal by the Information Industry Association to include the following language in Section 102(b):

However, copyright protection may exist in a collection of ideas or abstractions arbitrarily selected from a plurality of alternative ideas or abstractions or in a discretionary pattern of events or processes.

(Docket No. 345 at 8 (quoting Copyright Law Revision: Hearings Before the Subcomm. on Courts, Civil Liberties, and the Admin. of Justice of the House Comm. on the Judiciary, 94th Cong., 1st Sess. 334 (1975))). Because Congress did not

adopt this "exception," Borland contends that Congress manifested an intent that a "discretionary pattern of events or processes" is not protected under copyright law and should be determined to be a "system" under section 102(b).

Nothing Borland presents to the court, however, provides any explanation of why Congress did not include this language in the statute. Failure to adopt this language may manifest a Congressional belief that the language was unnecessary or irrelevant. Indeed, Congress did not adopt any statutory language denying copyright protection for "a collection of ideas or abstractions arbitrarily selected from a plurality of alternative ideas or abstractions or in a discretionary pattern of events or processes."

In any event, the explanatory language accompanying the proposal indicates that the proposal was intended to assure that copyright protection is accorded to nonliteral aspects of computer program code. Thus, Borland's argument may be interpreted as a contention that nonliteral aspects of computer code are not copyrightable, under section 102(b). If so, Borland's argument has been consistently rejected by the courts, including this one.

*See Paperback*, 740 F.Supp. at 54; Arthur R. Miller, Copyright Protection for Computer Programs, Databases, and Computer Generated Works: Is Anything New Since CONTU?, 106 *Harv.L.Rev.* 978, 994-96 nn. 78-88 (March 1993) (collecting cases). *See also Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 702 (2d Cir.1992) ("We have no reservation in joining the company of those courts that have already ascribed to this logic [that non-literal structures of computer programs are protected by copyright].").

I again reject the argument that copyright cannot extend to nonliteral aspects of a computer program. Moreover, Borland is wrong when it argues that, by failing to adopt the proposed statutory language, Congress manifested any intent concerning the definition of "system" in section 102(b).

Borland next argues that its phantom menus must constitute a "system," or be found "utilitarian" in nature, because the

phantom menus do not appear on the screen and are not communicated to the user.

This argument is readily dismissed. A user may never see computer code, but copyright protection is accorded to the code. Borland proposes a distinction on the ground that computer code may still be printed and read by someone. This distinction is vacuous for three reasons. First, the phantom menus may also be printed out; exhibit 517 is one copy. If one accepts the proposition that nonliteral aspects of computer code are copyrightable, the fact that the printed form of what Borland copied is not identical to any Lotus code or is not actually displayed to the user is not material to a finding that the Lotus menu structure contains copyrightable expression. Second, the fact that the phantom menus are not displayed does not mean that the user does not know they are there. Finally, copyright protection has been accorded to forms of computer code that are not generally intelligible to humans. *See Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983) (object code), *cert. dismissed*, 464 U.S. 1033, 104 S.Ct. 690, 79 L.Ed.2d 158 (1984).

Last, Borland argues that it has removed anything expressive in the Lotus menu tree from its phantom menus, leaving only the first letters of the command names as "markers" of the program's functionality. *See* Docket No. 311 at 25-26. This is not the case. The letters serve as "markers" of the Lotus menu tree structure. Claiming that the "markers" are part of the functionality again incorrectly assumes that the "idea" or "function" of Lotus 1-2-3 includes menu structure and first letters of the command names. Moreover, for the reasons explained in the Phase I Opinion, I conclude that the menu tree structure is original and was not dictated or even substantially limited by functional considerations at the time of its development. Accordingly, I also conclude that the menu structure and organization (including the first letter of the commands, used to mark the structure) are part of the protectable expression found in the Lotus 1-2-3 program.

In sum, I conclude that the Lotus menu structure, organization, and first letters of the command names are not within

the definition of the "idea," "system," "process," "procedure," or "method" of the Lotus 1-2-3 program and constitute part of the protectable expression found in the program.

## 2. *Borland's arguments concerning earlier "holdings" of this court.*

In arguing that the copying found in Borland's phantom menus is permissible, Borland places considerable reliance on what it characterizes as earlier "holdings of this court that expressly authorize a macro translator such as the Key Reader." *See* Docket No. 345 at 1-3. As a preliminary matter, before the Key Reader complaint in this case, no party before this court had ever pressed any actual dispute or controversy between two parties over whether a specific macro "translation" facility, standing alone, would infringe Lotus's copyrights. Article III courts decide only cases and controversies actually before the court. U.S. Const. Art III, § 2. Accordingly, I could not possibly have "held" in any earlier decision that macro translation as performed by the Key Reader is permissible under copyright law.

After examining Borland's specific citations to my earlier opinions, I also observe that in making this argument Borland has done a bit of interpretative twisting of the language of my earlier opinions. In earlier opinions, I have rejected a defendant's arguments that were based on the incorrect premise that the defendant was somehow required to copy specific elements of the Lotus 1-2-3 program.

*See Paperback*, 740 F.Supp. at 78 (no showing that copying of entire interface is necessary for macro translation); *Borland II*, 799 F.Supp. at 218 (distinguishing *Crume v. Pacific Mut. Life Ins. Co.*, 140 F.2d 182, 184-85 (7th Cir.), *cert. denied*, 322 U.S. 755[, 64 S.Ct. 1265, 88 L.Ed. 1584] (1944), as involving a system requiring use of only descriptive words, while in this case, varying full-word menu command names is not the only alternative for creating a different expression of the menu tree idea).

Contrary to Borland's arguments, I do not interpret the language in any of my earlier decisions rejecting these arguments as even remotely "authoriz[ing]" Borland to create its Key Reader. Explaining that a defendant's argument is overboard by use of an example is hardly equivalent to "holding" that the example is legally permissible. Moreover, in explaining my earlier decisions, I have twice observed that copying of expressive aspects of Lotus 1-2-3 may not be permissible under copyright law, even if that is the only way to achieve macro compatibility. See *Paperback*, 740 F.Supp. at 69; *Borland II*, 799 F.Supp. at 214. In *Borland II*, I stated that "I need not and do not decide whether Borland is prohibited from reading and interpreting macros that have been created by users of 1-2-3." 799 F.Supp. at 214. In short, I reject Borland's attempt, by selective reading and interpretation of language from earlier opinions, to twist discussions contained in those opinions into "holdings" of this court.

Finally, much of Borland's argument rests on statements concerning whether a "macro translator" such as in *Excel 2.1* infringes. Borland attempts to equate Key Reader with *Excel 2.1* by labeling the Key Reader as a "translator." As explained above, however, Borland has expanded the term "translator" from the one-time translation done in *Excel 2.1* to include on-the-fly interpretation as done in Borland's programs. In *Borland II*, I expressly did not decide issues concerning one-time translation, 799 F.Supp. at 214 (quoted *supra*), let alone issues concerning an expanded definition of macro "translator"—a definition including on-the-fly interpretation of macros.

#### **D. Substantial Similarity.**

For the following reasons, I find that what Borland copied into the Key Reader phantom menus is substantially similar to Lotus 1-2-3.

Contained in the phantom menus is a virtually identical copy of the menu structure and organization of Lotus 1-2-3, using the first letter of command names and other symbols to

delineate that structure. Although Borland's copy may use a different way of representing the menu structure than is contained within the Lotus code or displayed to the user while running Lotus 1-2-3, I find that what was copied by Borland (details of expression of the menu structure) is virtually identical to details of expression of the Lotus 1-2-3 program's menu structure.

I also conclude that differences in the method Borland uses to represent the menu tree structure in its phantom menus file does not negate a finding that the copied expression of the menu tree structure is substantially similar to the Lotus 1-2-3 program. First copyright law protects nonliteral aspects of a copyrighted program. Thus, one need not copy the specific code of a program to infringe copyrights in the program. Second, Borland's copying is analogous to a translation. No identity of words may exist in a translation into French of a book written in English. Nevertheless, the translated copy infringes by making a virtually identical copy of the book, but using a different method for representing the words and grammar structure. See 17 U.S.C. § 101 ("derivative work" includes translation); 17 U.S.C. 106(2) (copyright owner has exclusive right to prepare derivative works). Here, Borland created a virtually identical copy of the Lotus menu structure, but translated (nearly verbatim) the menu structure into a different language for representing menu structures.

Borland next contends that the fact that the phantom menus are not displayed on the screen prevents a finding of substantial similarity. I reject this contention for the same reason that I rejected Borland's argument that Borland copied only unprotected expression because the phantom menus are not displayed. See Section II.C.1, *supra*. That is, copyright law does not require that a program display its source or object code for copied elements of the code (literal or nonliteral) to be substantially similar.

In addition, I find that a user would understand that Borland was using a copy of the Lotus menu structure when executing a macro. Consider, for example, the "/{?}" macro described above. With this macro, the user can traverse the

Lotus menu structure precisely as though typing keys into the Lotus program. Similarly, Key Reader executes macros by sequentially parsing the (Lotus) macro text. This is demonstrated most clearly when one steps through macros in the Borland macro debugger. In addition, Borland's experts contend that a user may not even look to the full command names when using Lotus menus or macros. *E.g.* Liddle Dec'n, VI:11, ¶ 53. I accept this point and reject Borland's premise that the Borland programs do not demonstrate to the user that Borland copied Lotus's menu structure.

Finally, I find that the Lotus menu structure, including the first letters of the command names that mark that structure, constitutes a substantial part of the Lotus expression. What Borland copied into Key Reader is that portion of the Lotus program that determines precisely how the user may select from among the executable operations in the program. Qualitatively, Borland's phantom menus copy a substantial part of what a user would think of as constituting the Lotus program: the menu structure and the sequences of keystrokes for invoking each of Lotus's executable operations.

### III. Affirmative Defenses.

Borland raises four affirmative defenses: waiver, laches, estoppel, and fair use. Each is considered below.

#### A. Waiver.

The parties agree that in order to succeed in its waiver defense, Borland must prove that Lotus voluntarily and intentionally relinquished a known right. *See* Docket No. 311 at 26; Docket No. 338 at 20. That is, Borland must prove (1) voluntary and intentional relinquishment (2) of a right that Lotus was aware that it had.

Borland points to six statements that it contends demonstrate an intention to relinquish any right to assert claims against Key Reader. The first three statements were made before the Key Reader had been introduced into any Borland

product. For these statements, Lotus clearly could not have intended to waive claims specifically against Key Reader; Lotus was unaware of Key Reader. To overcome this obstacle, Borland contends that these statements reflect the intentional relinquishment of a known right to claim that *any* kind of macro "translator" infringes.

The three statements made before Key Reader was released were made by the president of and counsel for Lotus during proceedings in the *Paperback* case. *See* Manzi Dep'n, IX:41; Lemberg Dep'n, IX:24; Trial Transcript, VIII:B.16. Each of the statements concerns whether Lotus intended to assert a claim of infringement against macro translation as performed by Excel.

Borland first cites a statement by Jim Manzi, President of Lotus, made during a deposition in the *Paperback* case. The statement ends with the conclusion that "as far as I know, we have no issue of that [Excel] product." Entirely missing from this statement is any evidence (1) that Manzi was aware of a right to prevent copying found in Excel (even this assumes, without support, that (a) there was copying, and (b) Manzi was aware of that copying), or (2) that Manzi intended to waive a claim against products other than Excel, with respect to any type of macro translation. This statement does not reflect an intention to relinquish any known rights.

Borland next cites statements by Lotus's General Counsel, Thomas Lemberg. The Lemberg statements, in evidence under seal, were not made in a context associated with waiver of rights. Accordingly, I cannot find a manifested intention to relinquish known rights based on these statements. In addition, having examined the Lemberg testimony, I also find that Mr. Lemberg was referring to one-time translation of macros into a different macro language, rather than macro translation in general, and more specifically, rather than referring to any program that executes macros by reference to copies of Lotus's menus. Whether or not the distinction between one-time translation and direct execution of macros makes a material difference under copyright law is not significant; a party may waive rights without following the specific contours of

copyright law. In any event, I find that the Lemberg statements do not demonstrate an intent to relinquish either a specific right or more generally a right to assert claims against all forms of macro translation or execution.

Finally, Borland cites the *Paperback* trial transcript at a point where Lotus's counsel, Henry Gutman, responded to a defendant's contention that the defendant "had to" copy the entire Lotus user interface into its own spreadsheet product. Gutman argued that the premise of this argument is wrong because Excel had made a commercially successful product that did not copy the full interface, and in fact, included the ability to "translate[ ] and run in Excel" Lotus macros. VIII.B.16 at 45. Based on the materials before me, I find that (1) this statement was intended to discredit a defense argument and does not demonstrate that Lotus intended to waive any right, and (2) this does not demonstrate an intent to waive claims against all types of macro translation as opposed to Lotus's understanding of Excel's one-time translation.

Moreover, each of the three statements cited above was made in the context of an infringement suit against companies that had copied the entirety of Lotus's user interface. Up to the time of the three statements cited by Borland, Lotus had never been faced with issues raised by a program that did not fully copy the Lotus 1-2-3 interface but executed Lotus macros on-the-fly. Accordingly, for each of the statements I find that Borland has not proved (1) that, at the time of the statement, Lotus was aware of a right to bar copying of its menus when the copied menus were used only to execute macros on-the-fly, or (2) that Lotus intended to relinquish its right to assert claims against a program that directly executes Lotus macros by reference to copies of the Lotus menus, rather than by one-time translation of the macros into the product's own macro language.

Borland next cites three statements made by Lotus after Key Reader had been introduced into Borland's products. The first is a response to a Borland request for admissions. According to the response, Lotus admitted that

it does not contend that the . . . ability to execute "macros" originally written using Lotus 1-2-3 by means

of a conversion or translation program, standing alone, would infringe . . . , but qualifies its admission to state that Lotus does contend that the 1-2-3 User Interface of Quattro and Quattro Pro, including, *inter alia*, its use of the 1-2-3 menu structure to provide the ability to execute macros originally written using Lotus 1-2-3, impermissibly copies protected expression contained in Lotus 1-2-3.

See VII:B.1. Later in the same document containing Lotus's response to a request for admission, Lotus objected to a Borland request for an admission that the ability to execute Lotus macros while in Quattro Pro's native mode does not infringe. Lotus's objection was based on the ground that, as far as Lotus knew, Quattro Pro had no such ability. See VII:A.7.

This admission does not constitute a waiver of a right to assert infringement claims against Key Reader. Lotus explicitly stated that it *does* contend that use of the Lotus menu structure to provide the ability to execute macros is impermissible, as done in Borland's emulation interface. For the reasons explained in Section II.B above, I find that the Key Reader executes macros by using copies of the Lotus menu structure in a way very similar to the way that the 1-2-3 emulation interface executed macros. The qualification to the admission demonstrates an intent to pursue claims against a program doing precisely what Key Reader does.

Borland contends, however, that the reservation should be interpreted as objecting to a facility such as Key Reader only when it is a part of an emulation interface, and not when it is incorporated into Borland's native menu interface. Considering the circumstances of the admission, however, I find that such an interpretation is unreasonable. First, under Borland's proposed interpretation, the qualification to the admission does not qualify the scope of the admission. The admission without the qualification was already limited to macro translation "standing alone." Second, when specifically asked about running Lotus macros while using the native menu interface, Lotus's answer demonstrated that Lotus was not then aware of Borland's Key Reader feature. It is, therefore,

more reasonable to interpret Lotus's response as (1) a statement that Lotus was not contending in this suit that one-time translation or conversion infringes (a point very distinct from execution of Lotus macros by reference to "phantom" menus), or (2) a statement that Lotus was not contending that translation infringes because there was no need to make that claim in this case—Borland's emulation interface infringed and the scope of that infringement was sufficient for claims against the products then at issue.

If Borland wished later to be in a position to make its present assertion of waiver, it should have made full disclosure up front about Key Reader and obtained an unambiguous response. Borland's present claim of waiver is an attempt, wholly without merit, to escape the consequences of its own litigation strategy. Lotus was entirely free to respond as it did without waiving any right. I find that Lotus's response to the request for admission does not demonstrate an intent to relinquish a known right to assert claims against Key Reader, or a feature like Key Reader.

Borland next points to a statement by Lotus's counsel at a conference in June 1991. *See* VIII:B.17. In making the statement, Lotus's counsel advised the court that if Borland "include[d] 1-2-3 menus and alternate user interface" in a product Lotus believed was about to be released, Lotus would seek a preliminary injunction. Borland's contention that this constitutes a waiver of claims against any program that does not "display" the menus approaches, if not crosses, the line into frivolousness.

Finally, Borland points to a footnote in one of Lotus's briefs. *See* VIII:B.6. Here, Lotus was responding to Borland's contention that it *had* to copy and display the entire menu tree to achieve macro compatibility. In the footnote, Lotus argues that Borland was ignoring *this court's* distinction in *Paperback* between display of menus and executing macros by "interpretation, translation or conversion (as then existed in Excel or the foreign language versions of Lotus 1-2-3)." The foreign language versions of Lotus 1-2-3 perform a one-time translation or conversion of macros from English releases of

1-2-3 into a macro using the foreign language version of 1-2-3. *See* Morgan Dec'n, VI:12. This does not constitute waiver for several independent reasons.

First, the language focuses on the court's distinction, not Lotus's.

Second, Lotus sought to rebut Borland's argument by pointing out that macro compatibility might be achieved without display of the menus and cited examples of one-time macro conversion but not on-the-fly macro interpretation. In the circumstances here, I do not find that impeachment of a defendant's argument by example demonstrates an intent to relinquish any rights.

Third, at the time the argument was made, Borland had not released a product containing Key Reader but without the full emulation interface. Thus, Lotus was not faced with any decision over whether to assert claims against a program that copied Lotus's menus, but only for on-the-fly interpretation. I find that this statement does not demonstrate an intention to relinquish a known right to pursue infringement claims against a program containing Key Reader.

In sum, considering the statements proffered by Borland both separately and as a whole, I find that Borland has not proved that Lotus has waived a right to pursue its claims against Key Reader.

### ***B. Laches.***

To succeed in its laches defense, Borland must prove that (1) Lotus inexcusably or unreasonably delayed in raising claims based on Key Reader, and (2) Borland was prejudiced by this delay.

#### ***1. Delay.***

Lotus first raised claims of infringement based exclusively on Key Reader in its motion for leave to file supplemental complaint, filed December 7, 1992. (Docket No. 250.) Borland contends that this is more than two years after Key Reader was introduced. As explained above, however, Key

Reader was introduced into products that also contained the emulation interface (Quattro Pro 2.0, 3.0, and 4.0).

I begin by examining whether Lotus's delay was unreasonable or inexcusable during the time period when Borland's products included both the emulation interface and Key Reader. I then examine the delay in filing suit after Borland removed the emulation interface and marketed a product containing only Key Reader, *i.e.*, after Quattro Pro 4.01 was announced in August 1992.

For the following reasons, I find that the delay in filing suit before Borland introduced Quattro Pro 4.01 into the market was both reasonable and excusable.

First, even assuming Lotus was aware or should have been aware of Key Reader in Quattro Pro 2.0, 3.0, and 4.0, a fact disputed by Lotus, there is little reason that Lotus should have known (and no evidence that Lotus did know) that Key Reader was anything other than a link to the macro execution facility that is contained in the 1-2-3 emulation interface. In these circumstances, it would have been reasonable for Lotus to believe that if the emulation interface were removed Borland's Key Reader would not function. Indeed, Borland's Chief Executive Officer stated at a deposition in May 1991 that Quattro Pro would not be macro compatible with Lotus 1-2-3 if the 1-2-3 compatible menu tree were removed. *See* I:13 at 52. Thus, claims against the emulation interface might then have appeared to protect Lotus fully, without the added cost of asserting separate claims against Key Reader.

Second, before Quattro Pro 4.01 was released, Lotus was pursuing its strongest claims against all of Borland's existing spreadsheet products, *i.e.*, claims based on the emulation interface. In the circumstances of this case, I cannot say that Lotus was required to assert claims against each feature that Borland added to its (already) infringing products or be faced with the prospect of losing the right ever to assert claims against a program containing that feature. Until the emulation interface was removed, Lotus was not faced with a product that would not be found infringing without a determination that Key Reader infringes. I conclude that it was reasonable

to wait to assert claims against Key Reader until resolution of this dispute was necessary and could be raised explicitly as a live issue.

Third, to the extent Lotus was expressly pursuing claims against macro execution as performed by Borland's emulation interface, *i.e.*, on-the-fly macro interpretation by reference to Lotus's menu structure (*see* VII:B.1, quoted *supra*), Lotus was pursuing claims against something very similar to Key Reader.

Considering these factors, I find that Lotus's failure to bring an action specifically directed at Key Reader for the time period from the incorporation of Key Reader in Quattro Pro 2.0 until release of Quattro Pro 4.01 was both reasonable and excusable because of Lotus's pursuit of infringement claims against the emulation interface in each of Borland's then existing products.

Next, I consider whether Lotus's delay in filing suit after Quattro Pro 4.01 was released was unreasonable or inexcusable. Quattro Pro 4.01 was released after the *Borland II* decision was issued on July 31, 1992. Lotus filed its motion for leave to file supplemental complaint approximately four months after Quattro Pro 4.01 was made public. A few months before Quattro Pro 4.01 was released, Lotus gave notice of an intent to preserve claims against Key Reader (in April 1992, *see* VIII:A.5). Borland acknowledged this notice. (*See* VIII:A.2.) I also note the propriety of taking some time to investigate how Key Reader works before filing a complaint. In light of the brevity of the time period, the nature of investigation of Key Reader that was reasonably required, and Lotus's earlier notice of its intent to preserve claims against Key Reader, I find that Lotus did not unreasonably or inexcusably delay in bringing an action specifically directed at Key Reader.

Further, I reject Borland's contention that the motion to supplement the complaint was filed so late in the proceedings on the original complaint that Lotus's delay was unreasonable. It was the late (and secretive) timing of the release of Quattro Pro 4.01 that resulted in the late timing of Lotus's

claims. Moreover, Lotus agreed to a short time schedule for preparing and presenting its claims for both of the first two Phases of the trial. Accordingly, I find that Lotus's delay in filing claims against Key Reader was reasonable and excusable, despite the advanced stage of proceedings under the original complaint.

## 2. Prejudice.

As an example of alleged prejudice caused by Lotus's delay in bringing suit, Borland contends that an appeal of copyrightability issues to the First Circuit has been delayed. To make this claim, Borland contends that it removed the emulation interface (but not the Key Reader) in order to expedite an appeal to the First Circuit. The record is barren of evidence to support what counsel contends was Borland's motive for removing the emulation interface.

In addition, even assuming *arguendo* that this unsupported assertion is true, I find that Borland has not proved prejudice. For the reasons explained on the record at the September 23, 1992 conference, I denied Borland's motion for interlocutory appeal. This was before Lotus moved for leave to include claims against Key Reader in this case. Thus, before Lotus filed its motion for leave to supplement, I had already concluded that an interlocutory appeal was not appropriate. Borland has not demonstrated that it could have secured an interlocutory appeal of the summary judgment decision, even if Lotus never asserted claims against Key Reader.

To demonstrate prejudice, Borland also points to costs in developing and advertising Key Reader. Even assuming these costs were appreciable, a finding that Borland provides little evidence to support, I find that these costs were not incurred *as a result* of any delay by Lotus. Of course, Lotus had not delayed at all in bringing suit before Key Reader was introduced into a Borland product. In any event, as to Quattro Pro 2.0, 3.0, and 4.0, these products have been found to infringe whether or not Key Reader was present. Borland points to no costs with respect to versions 2.0, 3.0, and 4.0 that might have been avoided had Lotus asserted infringement claims directed

at Key Reader earlier. Moreover, Borland did not remove the emulation interface from Quattro Pro 2.0, 3.0, or 4.0, all released after Lotus had filed the complaint in this action. I find that Borland would have included Key Reader along with the emulation interface in its products whether or not Lotus immediately asserted that Key Reader infringes.

With respect to Quattro Pro 4.01 and later releases, Borland released these programs months after Lotus had sent a message (that Borland had acknowledged) that Lotus was seeking to preserve claims against programs containing Key Reader. (See discussion in preceding section.) I conclude that any prejudice suffered by Borland's inclusion of Key Reader in Quattro Pro 4.01 and later releases was not caused by delay on Lotus's part.

## C. Estoppel.

As was true with respect to the estoppel claims in Phase I, the parties apparently agree that to establish a defense of estoppel in relation to Phase II issues, Borland must prove that Lotus engaged in (1) conduct that induced Borland to change its position in good faith, or (2) conduct on which a reasonable person would rely. As I did in the Phase I Opinion, again without endorsing this specific formulation of the standard for deciding a claim of estoppel, I proceed, to Borland's advantage, to apply the standard stated by the parties.

In determining whether Borland has proved reasonable reliance on any conduct by Lotus, I consider first Borland's inclusion of Key Reader in Quattro Pro versions 2.0, 3.0, and 4.0, then turn to inclusion of Key Reader in Quattro Pro 4.01 and later releases.

Versions 2.0, 3.0, and 4.0 were released after Lotus filed suit claiming that Quattro and Quattro Pro infringe. Nevertheless, Borland included the emulation interface in each of these products. Accordingly I find that, even if Lotus had clearly and unequivocally stated an intention to assert claims against something like Key Reader, Borland would have

included Key Reader in these products anyway. Borland did not rely on any statement by Lotus in developing Key Reader.

For the following reasons, I also find that Borland has not proved that a reasonable person would have relied on any conduct by Lotus in developing or advertising Key Reader in versions 2.0, 3.0, and 4.0.

Borland contends first that it relied on Lotus's original complaint, ¶¶ 17-19, to conclude that Lotus would never assert claims against a program that copied Lotus's menu structure within the computer code but did not display the Lotus menu structure on the screen. This contention is without merit. The complaint alleged claims against products that did not have a Key Reader. The complaint focused on the emulation "user interface" because that is precisely the portion of the programs (in existence at the time the complaint was drafted), both as displayed on the screen and as implemented by the computer code, that infringes. In any event, without Borland's computer code, there is no Borland emulation interface. The complaint cannot be read in any way other than as asserting claims for infringement based on Borland's computer code. Finally, the premise to Borland's argument, that Lotus knew or should have known enough about the computer coding of Quattro and Quattro Pro to make specific allegations in the complaint about the computer code is without support in the record. Borland's claim that it reasonably relied on the original complaint as demonstrating an intention by Lotus to pursue infringement claims based only on Borland's display of the interface and not on Borland's code is baseless.

Borland next cites three statements, made after Key Reader was first released in Quattro Pro 2.0, by Lotus executives or counsel. These three statements are the same statements that Borland cited in its waiver argument, *i.e.*, (1) the admission, (2) the statement at the June 1991 conference, and (3) a footnote in one of Lotus's briefs. For the reasons explained in Section III.A, *supra* I conclude that Borland could not reasonably rely on any of these statements as demonstrating an intent not to pursue claims against Key Reader. Moreover, to

the extent that the statements and the definition of macro "translation" are ambiguous, reliance on those statements would be unreasonable. A reasonable person or company in Borland's position would have sought clarification before relying on the statements as a position adopted by Lotus.

In addition, each of these statements was made after Key Reader was introduced into Quattro Pro 2.0. Borland, when originally developing and introducing Key Reader, could not have relied on statements made later, after Key Reader had been developed. Nevertheless, Borland contends that it relied on these statements in retaining Key Reader in Quattro Pro 4.01 and later releases. As explained above, however, in April 1992, Lotus clearly stated (and Borland acknowledged) that Lotus wished to preserve claims against Key Reader. In light of this, I find that for Quattro Pro 4.01 and later releases, Borland could not have reasonably relied on the cited statements as demonstrating an intent by Lotus not to assert claims against Key Reader.

Finally, Borland's General Counsel avers that, when recommending to keep Key Reader in the program, he relied "most directly" on the court's language in the *Borland II* summary judgement opinion. See VI:9, ¶¶ 13-14. First, for the reasons explained in Section II.C.2, I find that Borland's selective reading of the opinion and twisted interpretation of the court's language is unreasonable; reliance on this language, therefore, also would be unreasonable. Second, I am not aware of any precedent for basing an estoppel defense not on conduct of the party alleged to be estopped, but on conduct of the court. Borland offers no authority to support this proposition. Instead, Borland contends that the court's language reflects Lotus's arguments. I reject this contention. To support such a contention, Borland would have to place in evidence Lotus's arguments and conduct, rather than simply pointing to the court's decision. I have examined the language Borland points to in Lotus's earlier briefs and argument and reject Borland's contention that Borland reasonably relied on Lotus's statements or conduct.

In any event, to the extent Borland relied on the court's language in deciding to include Key Reader in Quattro Pro 4.01, Borland was not relying on Lotus's conduct.

Having considered the several instances of conduct cited by Borland both individually and as a whole, including any delay in filing suit, I find that Borland has not proved its estoppel defense.

#### **D. Fair Use.**

Borland contends that copying of the Lotus menu structure and first letters of the menu commands is a fair use of Lotus's copyrighted programs. The fair use doctrine is an "equitable rule of reason." *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 448, 104 S.Ct. 774, 792, 78 L.Ed.2d 574 (1984) (quoting legislative history). The doctrine

permits courts to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which that law is designed to foster.

*Stewart v. Abend*, 495 U.S. 207, 236, 110 S.Ct. 1750, 1768, 109 L.Ed.2d 184 (1990) (quoting *Iowa State University Research Found. Inc. v. American Broadcasting Cos.*, 621 F.2d 57, 60 (2d Cir.1980)).

Congress set forth four statutory factors to guide consideration of what constitutes a fair use. Section 107 of the Copyright Act provides that:

the fair use of a copyrighted work . . . for purposes such as criticism, comment, news reporting, teaching . . . , scholarship, or research, is not an infringement of copyright. In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include—

- (1) the purpose and character of the use, including whether such use is of a commercial nature or is for non-profit educational purposes;
- (2) the nature of the copyrighted work;

(3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and

(4) the effect of the use upon the potential market for or value of the copyrighted work.

17 U.S.C. § 107. These factors are not exclusive. Each case is to be decided on its own facts. *See Harper & Row, Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 560, 105 S.Ct. 2218, 2230, 85 L.Ed.2d 588 (1985).

#### **1. "[P]urpose and character of the use."**

The first statutory factor focuses primarily on whether the copied materials are used to secure profit.

The crux of the profit/nonprofit distinction is not whether the sole motive of the use is monetary gain but whether the user stands to profit from exploitation of the copyrighted material without paying the customary price.

*Harper & Row*, 471 U.S. at 562, 105 S.Ct. at 2231. Section 107(1) contrasts a use that is "of a commercial nature" with a use that is for "nonprofit educational purposes." According to the preamble in section 107, copying for "criticism, comment, news reporting, teaching . . . , scholarship, or research may be closer to being a "nonprofit educational use" than a commercial use.

In the instant case, there can be little dispute that Borland copied from Lotus for "commercial" purposes. Borland sells its spreadsheet products for profit. The Key Reader is included to make the program more attractive to potential customers than have Lotus 1-2-3 macros. Borland has copied Lotus's menu structure in order to enhance profits from sales of its spreadsheet products, without having secured a license to do so.

Borland tries to minimize the impact of having a profit motive on its fair use defense by contending that Key Reader is actually used by consumers and only for interpreting the macros that the consumers have written. (Docket 345 at 19.) The consumer's use of the part of Quattro Pro that Borland

copied from Lotus, however, does not change the character of *Borland's* use of the copied materials—to please consumers and increase sales. To contend that copying is closer to a fair use because customers want the copied materials is entirely without merit. “It is fundamentally at odds with the scheme of copyright to accord lesser rights in those works that are of the greatest importance to the public.” *Harper & Row*, 471 U.S. at 559, 105 S.Ct. at 2230.

Borland nevertheless contends that the present case is “almost identical” to *Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc.*, 964 F.2d 965 (9th Cir.1992), *cert. denied*, \_\_\_ U.S. \_\_\_, 113 S.Ct. 1582, 123 L.Ed.2d 149 (1993). In *Galoob*, the alleged infringer sold a product (the “Game Genie”) that allowed private users of Nintendo’s copyrighted games to use Nintendo’s games in an enhanced fashion. Nintendo did not argue that the Game Genie directly infringes Nintendo’s copyright. Rather, Nintendo argued that use of the Game Genie with a Nintendo game creates a derivative work of the copyrighted Nintendo game. *See id.* at 970. Thus, Nintendo’s infringement claims concerned only whether Galoob contributed to a consumer’s direct infringement of Nintendo’s copyright when the consumer used the Game Genie with a Nintendo game at home. For this reason, the opinion concerning fair use explicitly restricted its focus to whether a consumer’s private use of the Game Genie was a fair use (and not on Galoob’s use of the Game Genie—sale to the consumer for profit). *Id.* Private use by a consumer is a nonprofit use rather than a commercial use. *See id.* *See also Sony Corp.*, 464 U.S. at 449, 104 S.Ct. at 792.

The present case is readily distinguished from *Galoob*. Lotus contends and I have found that Borland copied and that the Quattro Pro products, including Key Reader, directly infringe Lotus’s copyright. The fair use issues in this case properly focus on Borland’s use (and not the consumer’s use) of the infringing products. Otherwise, a book store could copy and sell “Gone With the Wind” without permission. The book store would rely on the consumer’s private use of the copied material to claim that its copying is a fair use.

Borland cites *Sega Enters. Ltd v. Accolade, Inc.*, 977 F.2d 1510, 1522 (9th Cir.1992) for the proposition that “the commercial nature of a use is a matter of degree, not an absolute.” In finding that the commercial nature of the defendant’s use in *Sega* was of a lesser degree, the Ninth Circuit relied on the fact that the copied materials were not included in defendant’s final product. *Id.* Here, however, what Borland copied from Lotus is included in the product it sells to the public.

Borland also relies on *Sega* to argue that Borland has performed its own creative work in developing Quattro Pro, and therefore, Key Reader merely “supplements” Borland’s own creative work. Assuming this to be true does not change the character of Borland’s commercial use of the copied material. Most of Borland’s own creative work, including its own menu structure, exists in Borland’s programs independently of whether or not Borland includes a Key Reader in its product. Borland used the copied materials to please customers and increase sales, independently of the creative work it did. In addition, the *Sega* case is readily distinguished. *Sega* involved “intermediate” copying; the defendant did not include copied materials in the final product. 977 F.2d at 1522. Here, Borland “supplemented” the product it sells to consumers with a facility including the impermissibly copied materials.

In sum, I find that the “purpose and character” of Borland’s use of Key Reader is entirely commercial. This conclusion concerns just a single factor among a number of factors a court weighs in deciding a fair use defense. Nevertheless, when a defendant’s use of copied materials is determined to be commercial, that use is “presumptively an unfair exploitation of the monopoly privilege that belongs to the owner of the copyright.” *Abend*, 495 U.S. at 237, 110 S.Ct. at 1768 (quoting *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793). *See also Harper & Row*, 471 U.S. at 562, 105 S.Ct. at 2231.

## 2. “[T]he nature of the copyrighted work.”

This statutory factor focuses on locating a work, according to its nature, on the spectrum from factual works to fiction or fantasy. “The law generally recognizes a greater need to

disseminate factual works than works of fiction or fantasy." *Harper & Row*, 471 U.S. at 563, 105 S.Ct. at 2232. Similarly, a work largely dictated by functional considerations is closer to the "factual works" end of the spectrum than the fiction end. *See Sega*, 977 F.2d at 1524.

Also, even for factual works,

there are gradations as to the relative proportion of fact and fantasy. One may move from sparsely embellished maps and directories to elegantly written biography. The extent to which one must permit expressive language to be copied, in order to assure dissemination of the underlying facts, will thus vary from case to case.

*Harper & Row*, 471 U.S. at 563, 105 S.Ct. at 2232 (quoting Gorman, *Fact or Fancy? The Implications for Copyright*, 29 J. Copyright Soc. 560, 561 (1982)). Thus, weighing this factor requires a determination of how much freedom of expression the author had in formulating the copyrighted expression. If there was little freedom, the expression is like a bare map. With greater expression found in the work, the scope of copyright protection is greater and a finding of fair use less likely.

For the reasons explained in the Phase I Opinion, I concluded that Lotus had substantial freedom of expression in forming its menu tree. The degree of freedom of expression in creating a menu structure using only first letters of command names is no less than the degree of freedom in designing Lotus's menu tree using full command names. Nevertheless, the menu tree is based on the set of executable operations selected for Lotus 1-2-3. Accordingly, I find that the copyrighted work in this case is similar to a "factual work," but that copying from the substantial expression found in the menu tree is not remotely necessary for disseminating the underlying executable operations (or "facts"). I find that this factor does not weigh significantly in either party's favor.

Borland contends that the "utilitarian nature" of the menu tree weighs in favor of a finding of fair use. Lotus's copyright does not, however, extend to aspects of the program that are utilitarian or functional and not expressive. Moreover, Bor-

land's reliance on *Sega* is misplaced. The *Sega* court determined that copying of protected expression was necessary in order to gain access to unprotected aspects of the program. For this reason, the *Sega* court determined that copyright protection for the code in *Sega* was entitled to a lower degree of protection than other literary works (*i.e.*, less protection against the "intermediate" copying done by the defendant in that case). 977 F.2d at 1526. Here, Borland's copying was not necessary to examine unprotected aspects of Lotus 1-2-3.

**3. "[A]mount and substantiality of the portion used in relation to the copyrighted work as a whole."**

Borland created a virtually identical copy of the menu structure and first letters of command names from Lotus 1-2-3's menu tree. This necessarily includes copying of the keystroke sequences used by Lotus to access executable commands and Lotus's macro language. For the reasons explained in Section II.D, *supra*, I find that this is a substantial part of the copyrighted expression in Lotus 1-2-3.

Borland argues that "it is difficult to see how Borland could use even less of 1-2-3." Docket No. 345 at 21-22. The answer to this is simple. Borland could have copied none of Lotus's expression, or (though I do not comment on the permissibility of doing so) less than the complete 1-2-3 menu structure and first letters of command names.

The fact that Borland uses the menu tree structure and first letters of the command names only for macro translation is not significant in evaluating this statutory factor. "[A] taking may not be excused merely because it is insubstantial with respect to the *infringing* work." *Harper & Row*, 471 U.S. at 565, 105 S.Ct. at 2233 (original emphasis).

I find that this statutory factor weighs in favor of Lotus.

**4. "[T]he effect of the use upon the potential market for or value of the copyrighted work."**

The fourth statutory factor focuses on whether "some meaningful likelihood of future harm exists." *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793 (original emphasis). To negate a claim

of fair use, a copyright holder may show either "that the particular use is harmful, or that if it should become widespread, it would adversely affect the potential market for the copyrighted work." *Id.* (holding that, for a noncommercial use, copyright holder must make this showing). "This inquiry must take account not only of harm to the original but also of harm to the market for derivative works." *Harper & Row*, 471 U.S. at 568, 105 S.Ct. at 2234.

As noted in Section III.D.1, *supra*, a commercial use of the copied material is presumptively unfair. Thus, "[i]f the intended use is for commercial gain, [the] likelihood [of future harm] may be presumed." *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793.

In the present case, I find that there is a meaningful likelihood that Borland's copying negatively affects the market for and value of Lotus's copyright. Quattro Pro's functionality is largely equivalent to Lotus 1-2-3's functionality. The operations the programs perform on spreadsheets are similar in nature. From this, I infer that many users will not go to the expense of purchasing, maintaining, and updating both the Quattro Pro and Lotus lines of spreadsheet products.

Borland's inclusion of Key Reader caters to users who already own Lotus 1-2-3; *i.e.*, users that have developed macros for use with Lotus 1-2-3. This may directly affect sales of Lotus 1-2-3 programs. For example, a company using Lotus 1-2-3 may decide to shift to use of Quattro Pro rather than purchasing additional copies of Lotus 1-2-3 for new employees. Inclusion of Key Reader may also have an impact on sales of updates to or new releases of Lotus 1-2-3. For example, Key Reader increases the likelihood that a user will shift to Quattro Pro rather than purchase an update to Lotus 1-2-3.

For these reasons, I infer that there is a meaningful likelihood of harm to Lotus (by a negative impact on the market and a decrease in value of Lotus's copyright) stemming from Borland's use of Key Reader.

Borland seeks to rebut this inference with evidence that Lotus's share of the spreadsheet market has remained steady

from the time before Borland introduced Key Reader to the present. I reject this argument for the following reasons.

First, this statutory factor includes an examination of what would happen "if [the use] should become widespread." *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793. This is a necessary consideration. Otherwise, a local bookstore would be free under the fair use doctrine to copy and sell a popular book (without license) because the impact on the national market is negligible. Here, if consumers widely purchase Quattro Pro intending to use Key Reader, I find that the market for Lotus 1-2-3 products and value of the Lotus copyright would be diminished.

Second, Borland provides no evidence that maintaining market share is equivalent to having a copyright of undiminished value. A number of factors other than a complete lack of impact on the market for Lotus 1-2-3 may explain a constant market share. For example, pricing of both 1-2-3 and competing products (including cut prices or lack of increase in prices of 1-2-3) may maintain a constant market share. Also, advertising might be increased to maintain market share. The evidence presented by Borland falls far short of supporting an inference that Lotus market has not been adversely affected.

Third, even if Lotus can maintain a constant market share without increased costs associated with lower prices or higher expenses, this does not demonstrate that the copying found in Key Reader has not diminished Lotus's market share from what it would be if Borland had not copied. Without Borland's copying, Lotus's market share may have increased.

In short, I find that Borland's limited evidence concerning market share is entirely insufficient to demonstrate that Lotus has not been harmed or to rebut the presumption that Borland's commercial use of Key Reader presents a meaningful likelihood that Lotus will be harmed. This statutory factor weighs in favor of Lotus.

### 5. Other factors.

The statutory list of factors a court may consider in deciding the merits of a fair use defense is not exclusive. As an additional factor, Borland contends that public policy dictates that Borland's use of Key Reader be considered a fair use. Borland argues that permitting Borland to market an "improved" spreadsheet product while maintaining macro compatibility for users of Lotus 1-2-3 stimulates (Borland's) artistic creativity. In essence, Borland contends that, when forming new artistic expressions, Borland should be permitted to tap into the market for existing expression by copying not just the idea but by copying from the expression of the idea. I reject Borland's arguments for the following reasons.

First, this "new" factor is just another argument concerning the "purpose and character" of Borland's use of the copied expression. Thus, I reject Borland's argument for the same reasons that I concluded that Borland's use of the copied materials is commercial. Borland is not entitled to rely on the consumers private use of the program for execution of macros to claim fair use.

Second, even when barred from copying expression, Borland was and is free to create new expressions of the ideas contained in Lotus 1-2-3. Borland presents no evidence that providing users with macro compatibility, or the ability to execute macros by reference to copies of Lotus's menus, is necessary to permit copying of the ideas found in Lotus 1-2-3. *Compare with Sega*, discussed *supra*. Borland has not demonstrated that allowing it to copy Lotus's expression would stimulate creativity more than the general scheme of providing copyright protection for an author's expression.

### 6. Summary.

Each of the factors to be weighed in deciding whether Borland's copying constitutes a fair use either weighs against a finding of fair use, or is entitled to little or no weight. Considering all factors, I find that Borland's copying is not a fair use of Lotus's copyrighted expression.

## IV. Summary.

For the reasons explained above, I find that Borland's Key Reader infringes Lotus's copyrights. Further, I find that (1) Lotus has not waived claims against Key Reader, 2) Lotus is not barred by laches or estoppel from asserting claims against Key Reader, and (3) including expression copied from Lotus 1-2-3 in Borland's Key Reader facility is not a fair use of the protected expression.

### Permanent Injunction

For the reasons stated in the July 31, 1992 Memorandum and Order, and for the reasons stated and on the Findings and Conclusions recited in the Opinion of June 30, 1993, as amended August 19, 1993 (the "Phase I Opinion"), and the Opinion of August 12, 1993, as amended August 19, 1993 (the "Phase II Opinion"), it is ORDERED, ADJUDGED, AND DECREED:

1. Defendant Borland International, Incorporated ("Borland") has infringed plaintiff's copyrights in the computer software program "Lotus 1-2-3" by its development, manufacture and sale of: (1) Quattro and Quattro Pro version 1.0, by Borland's inclusion of its "1-2-3 emulation" interface; (2) Quattro Pro versions 2.0, 3.0, and 4.0, by Borland's inclusion of its "1-2-3 emulation" interface and its "Key Reader" facility; and (3) Quattro Pro SE, 4.01 and Quattro Pro for Windows by Borland's inclusion of its "Key Reader" facility.

2. Acting under 17 U.S.C. § 502(a), the court orders that Borland, its officers, agents, servants, employees and attorneys, and all persons in active concert or participation with them, are hereby permanently enjoined from manufacturing, selling, distributing, licensing, or continuing to license for manufacture, sale, distribution or sub-licensing

(i) Quattro, Quattro Pro versions 1.0, 2.0, 3.0, 4.0, SE, 4.01, and all versions of Quattro Pro for Windows that

have been or are on the market on this date (August 19, 1993) ("Quattro Pro for Windows"), and

(ii) any work that contains in any portion, component or module thereof, a copy of the Lotus 1-2-3 menu commands and/or menu structure, in any form. Borland may at any time, however, apply to this court for modification of this clause upon a showing of good cause for determining that Borland has developed or proposes to develop a product that is within this clause but for special reasons

(a) the product is not infringing, or

(b) an alternative form of remedy allowing marketing on specified conditions, including payment of royalties, is more appropriate than an injunction.

3. In addition, upon entry of this Permanent Injunction, Borland shall forthwith take all necessary steps to terminate any existing licenses, distribution agreements or other arrangements pursuant to which it has manufactured, sold, distributed or licensed any of Quattro, Quattro Pro version 1.0, 2.0, 3.0, 4.0, SE, 4.01 or Quattro Pro for Windows and it shall, at its own expense, on or before August 27, 1993, notify each and every other party to any such license, distribution agreement or other arrangement of the existence and terms of this Permanent Injunction. Borland shall take all reasonable measures necessary to ensure, so far as it can control, that all such manufacturers, distributors and resellers comply with the terms hereof, including reasonable measures to prevent the selling of infringing products, which may include, but does not require, repurchase of products, and which does not include a general recall.

UNITED STATES DISTRICT COURT  
D. MASSACHUSETTS

Civ. A. No. 90-11662-K

June 30, 1993

As Amended Aug. 19, 1993

---

LOTUS DEVELOPMENT CORPORATION,

*Plaintiff,*

—v.—

BORLAND INTERNATIONAL, INC.,

*Defendant.*

---

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, MA, Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, MA, Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, CA, David L. Hayes, Mitchell Zimmerman, Fenwick & West, Palo Alto, CA, Peter E. Gelhaar, Donnelly, Conroy & Gelhaar, Boston, MA, Gary L. Reback, Nina F. Locker, Peter N. Detkin, David A. Priebe, James A. DiBoise, Wilson, Sonsini, Goodrich & Rosati, Palo Alto, CA, Laurence H. Tribe, Cambridge, MA, *for defendant.*

---

## OPINION

KEETON, *District Judge*.

On July 31, 1992, the court allowed, in part, a motion for summary judgment filed by plaintiff Lotus Development Corporation ("Lotus") and denied the cross motion for summary judgment by defendant Borland International, Incorporated ("Borland"). After extended procedural maneuvering, the parties agreed to try remaining liability issues without a jury. An explanation of the proceedings leading up to the trial is essential to precise identification of the issues raised by the parties in the nonjury trial of February 1-3 and March 31-April 2, 1993.

### I. Earlier Proceedings.

#### A. Partial Summary Judgment.

This Opinion assumes the reader's familiarity with the Memorandum and Order allowing, in part, Lotus's motion for summary judgment. That document was published as *Lotus Dev. Corp. v. Borland Int'l Inc.*, 799 F.Supp. 203 (D.Mass.1992) (July 31 Memorandum and Order). In addition, the terminology used in this Opinion follows the terminology set forth in detail in the earlier Memorandum and Order. *Id.* at 206-208. Background information appears in two earlier documents issued by this court. The first is a Memorandum and Order in this case, *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F.Supp. 78 (D.Mass.1992). The second is an opinion in a related case involving claims of infringement of copyrights for the Lotus 1-2-3 program. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass.1990).

The July 31 Memorandum and Order explained the standard to be applied in this case for determining copyrightability issues:

FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives

that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation, some conception of the "idea," "system," "process," "procedure," or "method"—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

\* \* \*

SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work."

*Borland*, 799 F.Supp. at 211 (quoting *Borland*, 788 F.Supp. at 90 (quoting *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37, 60-61 (D.Mass.1990)) (all emphases omitted)).

Applying this test, I determined that the "idea," "system," "process," "procedure," or "method" of the Lotus 1-2-3 program is a menu-driven electronic spreadsheet whose

user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are

accessible through the paths of the menu command hierarchy.

*Id* at 216-17. (The meanings of menu command, menu structure and menu tree are explained at greater length below.) I concluded also that

as a matter of law, Borland's Quattro products infringe the Lotus 1-2-3 copyright because of (1) the extent of copying of the "menu commands" and "menu structure" that is not *genuinely* disputed in this case, (2) the extent to which the copied elements of the "menu commands" and "menu structure" contain expressive aspects separable from the functions of the "menu commands" and "menu structure," and (3) the scope of those copied expressive aspects as an integral part of Lotus 1-2-3.

*Id* at 223 (original emphasis).

Nevertheless, I concluded that determining the scope of relief in this case depends on resolution of disputed factual contentions because Lotus contended and Borland disputed "that the copying of separable expressive elements of the Lotus 1-2-3 user interface into the Quattro programs was greater than the minimum essential to constituting a substantial part of the Lotus 1-2-3 work," which I had determined on motion for summary judgment not to be genuinely in dispute. *Id*. In other words, there is no genuine dispute of fact that the Quattro and Quattro Pro programs infringe, but fact issues remain as to the scope of impermissible copying. Specific fact issues apparent on the record at that time concerned (1) whether Borland copied the long prompts of Lotus 1-2-3, (2) whether the long prompts contain expressive elements, and (3) the extent (if any) that functional constraints limit the number of possible ways that the Lotus menu command hierarchy could have been arranged at the time of its creation. *See* Order Regulating Jury Trial, September 30, 1992 (Docket No. 232) at 20.

In addition, I concluded that Lotus was entitled to summary judgment against Borland on the affirmative defense of

waiver, but not on the affirmative defenses of laches and estoppel. *See Borland*, 799 F.Supp. at 222-23.

## B. Further Proceedings Before Trial.

Up to the time of the court's ruling of July 31, 1992, the parties' contentions concerned issues raised in the allegations of the "original complaint" filed July 2, 1990, concerning infringement of Lotus 1-2-3 by Quattro and Quattro Pro's "emulation interface." Trial for the remaining liability issues in the original complaint was scheduled before a jury. In January 1993, this court permitted Lotus to file a supplemental complaint alleging copyright infringement by Borland in programs containing a "Key Reader" feature (which is described in some detail in the August 12, 1993 Opinion that addresses the issues of the "Key Reader" phase of the trial)—specifically in Quattro Pro versions 2.0, 3.0, 4.0, SE, 4.01 and Quattro Pro for Windows. *See* Docket 250, Exh. A.

After these developments and as the parties were preparing for trial of issues raised in the original complaint, the parties entered into a series of stipulations that altered the nature of the proceedings. *See* Stipulation and Order Regulating Trial (Docket No. 330); Stipulation and Order Regulating Key Reader Trial (Docket No. 349).

The first set of stipulations concerned trial of issues raised in the original complaint. These stipulations

govern the trial of all issues not previously finally decided by way of summary judgment concerning Borland's alleged liability herein, and all its defenses thereto, *excluding* the issues raised by Lotus' Supplemental Complaint concerning the "Key Reader" feature (the "Phase I Trial").

Docket No. 330, ¶ 1. With respect to issues raised in the original complaint, the parties waived jury trial for the liability issues that had previously been scheduled for the Phase I trial.

With respect to the long prompts, the parties stipulated that:

1. The order of display of the long prompts within the "1-2-3-compatible" modes of Quattro and Quattro Pro

follows the order of display of the menu commands within those modes, and each such long prompt provides a short textual description of the command to which it relates.

2. Lotus shall not contend, in this action or any appeal therefrom, that Borland has copied the long prompts of Lotus 1-2-3 in Quattro or Quattro Pro.

3. Borland shall not contend, in this action or any appeal therefrom, that Borland has not copied the long prompts of Lotus 1-2-3 in Quattro or Quattro Pro.

4. Neither party shall contend, in this action or any appeal therefrom, that the issue of whether or not Borland copied the long prompts of Lotus 1-2-3 in either Quattro or Quattro Pro is material to any other issue that has been or will be resolved in this case.

Docket No. 330, Exh. A, ¶¶ 1-4.

The second set of stipulations "govern[s] the trial of all liability issues (including any defenses thereto) raised by Lotus' Supplemental Complaint concerning the 'Key Reader' feature (the 'Phase II Trial')." Docket No. 349, ¶ 1. For trial of the liability issues raised by the supplemental complaint, the parties waived their rights to trial by jury. *Id.*

### C. Summary of Issues Before the Court.

Phase I of the trial was held on February 1-3, 1993. At that time, the issues before the court were the scope of infringement by Borland and Borland's affirmative defenses of laches and estoppel (the affirmative defense of waiver having been resolved at summary judgment). After the close of Borland's evidence, however, Borland was allowed leave to amend its answer to assert an affirmative defense of fair use. *See* Memorandum and Order, March 30, 1993 (Docket No. 353). In response, Lotus moved for judgment on partial findings. *See* Fed.R.Civ.P. 52(c). After hearing in open court and for the reasons stated on the record, I allowed Lotus's motion for

judgment on the issue of Borland's fair use defense to the original complaint (Phase I).

Phase II of the liability trial was held on March 31-April 2, 1993. The issues presented to the court included the full range of liability determinations for the "Key Reader" supplemental complaint. In addition, the parties tried Borland's defenses (to the "Key Reader" complaint) of waiver, laches, estoppel, and fair use.

This opinion addresses the issues raised in Phase I of the trial. On August 12, 1993, the court released an Opinion resolving the issues raised in Phase II.

### II. Scope of Infringement in Phase I Trial.

As I understand the parties' stipulations and arguments, Lotus does not now contend that Borland copied the entire 1-2-3 interface. Rather, Lotus claims that Borland has illegally copied the Lotus 1-2-3 "menu commands" and "menu structure." Accordingly, the only issues before the court concern copying of the menu commands and structure.

Borland contends that copying of menu commands and structure is permissible because of functional constraints on formulation of the menu commands and structure. To the extent that Borland contends that the menu commands and structure as a whole are not copyrightable, Borland's contention was rejected as a matter of law at summary judgment. Nothing Borland presented at the Phase I trial alters my view that there is no genuine dispute that "a large part of the structure and arrangement of the menu commands is not driven entirely by functional considerations." 799 F.Supp. at 218.

Although I determined that no genuine dispute of fact had to be resolved in order to determine that the menu commands and structure contain protectable expression, I also determined that disputed factual contentions might have to be resolved to determine the *scope* of infringement. Among factors bearing on the scope of infringement are (1) the scope of copying, and (2) the nature of the copied work. Given the

implications of the idea and the functional considerations, what is the extent of the expression? If there is "essentially only one way to express an idea," complete copying is permissible. *Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 606 (1st Cir.1988). If there are "only a limited number of ways of expressing the idea," then proof of "near identity" of copied expression is required to show infringement. *Id.* For a work "embodying only one of an infinite variety of ways of expressing an idea," duplication or near identity is not required. *Id.* at 607. Consequently, there is a "sliding scale" that determines the scope of copyright protection. Accordingly, I first examine more closely the scope of copying and then examine how much of this copying was impermissible.

#### A. Menu Commands and Structure—Scope of Copying.

##### 1. Defining and describing "menu commands and menu structure."

Omitting details not relevant for resolution of any dispute in this case, one may describe the idea of menu commands and menu structure in the following way.

Each spreadsheet program described in the evidence before the court has a set of basic executable operations that a user may invoke. Each executable operation does something with data in a spreadsheet (*e.g.*, erases data in a spreadsheet cell), fixes parameters of a spreadsheet (*e.g.*, width of a column in a given spreadsheet), fixes parameters of the program (*e.g.*, hardware configuration or default settings), or performs some other function such as printing or saving a spreadsheet.

For each spreadsheet program before the court, the number of executable operations is large. Also, the possible methods of presenting available operations to the user is large. One simple approach would be to give each operation a unique name. If this were done, however, the large number of unique names would make it difficult for a user to remember and invoke them expeditiously.

A common way of overcoming this problem is to develop a menu hierarchy. The first level of the hierarchy presents the user with a "menu" consisting of a limited number of "menu commands." Some of these menu commands may be executable operations (*e.g.*, "Quit") causing the program to do something (*i.e.*, terminate). When the user invokes an executable command, the program performs the corresponding operation and does not present any further menu options. To proceed farther in the hierarchy, the user must start again and select a different option.

Other menu commands are not in the set of executable operations; instead they are "internal" menu commands, each of which substitutes a new menu in place of the menu in which the internal command appears. Like the menu it replaces, the substituted menu (or "submenu") may consist of a combination of executable commands and internal commands.

The menu commands and menu structure are commonly described as a "tree." The imagery is imperfect, however, unless one thinks of a rather unusual tree that has a leaf or two as well as branches at most junctures. In this imagery, some of the choices at each juncture may be branches and others may be leaves. Each branch or leaf has a name. The "name" of the branch or leaf is a "menu command" within the program's "menu tree." The user starts at the trunk, and by choosing a branch, starts a climb upward. (One may, of course, also envision an upside-down tree, with the user working downward.)

If instead of choosing a branch at the first juncture the user chooses an executable operation, no climb occurs. A name (menu command) corresponding to an executable operation is a leaf of the tree. Having reached that leaf, the user cannot climb farther; no new branches are presented. Instead, the operation is executed. To climb farther (or elsewhere in the tree), the user must go back at least a bit (or all the way to the beginning) and choose branches up to a higher level. As long as the user invokes an internal menu command (a branch) at each juncture, the program presents to the user a new set of



refer to the menu tree generically as the Lotus 1-2-3 menu tree although I have examined both versions 2.0 and 2.01.

The Lotus 1-2-3 menu tree contains approximately 469 menu commands. *See* Kieras Dec'n, V:& at ¶ 66.

(The record in this case consists of nine volumes of material. Citations to this record are formatted as Volume:Tab; thus "V:8" means Volume V, Tab 8.)

Portions of the menu tree are illustrated at various places in the Exhibits. The entire tree is set out in Exhibit 524, A-C, E (the "Flesher Exhibits" which were originally submitted with the summary judgment materials). As the number of commands suggests, the menu tree is quite large.

The Quattro and Quattro Pro programs have both a native menu tree and a "1-2-3 emulation" menu tree. I do not understand Lotus to contend that the native menu tree infringes any Lotus copyright. Accordingly, when I refer to the Quattro or Quattro Pro menu trees, I refer only to the menu tree used in the program's 1-2-3 emulation interface.

The Quattro menu tree contains approximately 3370 menu commands. (Kieras Dec'n, V:8 at ¶ 77.) This menu tree is illustrated in Flesher Exhibit B. The Quattro Pro version 1.0 menu tree contains approximately 5216 commands (*Id.* at ¶ 66.) This menu tree is illustrated in Flesher Exhibit A.

The Quattro Pro program has been released in several versions, specifically versions 1.0, 2.0, 3.0, 4.0, 4.01, SE, and Quattro Pro for Windows. Only versions 1.0, 2.0, 3.0, and 4.0 contain the emulation interface. In deciding issues for Phase I of the trial, therefore, when I refer, without further specification, to the Quattro Pro menu tree, I refer only to the emulation menu tree in Quattro Pro versions 1.0, 2.0, 3.0, and 4.0.

The entire Lotus menu tree is contained within both the Quattro and Quattro Pro version 1.0 menu trees. For example, the first menu in Quattro Pro version 1.0 contains an identical copy of the 1-2-3 menu commands, in the same order, but with one new command inserted ("View"). Invoking a Borland command that is identical to a Lotus command produces a menu that is an identical copy of the Lotus submenu, but (in

some cases) with one or two new commands inserted. Thus, each menu or submenu in Lotus 1-2-3 is reproduced identically, but with the insertion in the Quattro and Quattro Pro menu trees of some new menu commands and any submenus associated with the new menu commands.

Put another way, both the Quattro and Quattro Pro version 1.0 menu trees consist of a *virtually identical* copy of the entire 1-2-3 menu tree, with new branches or leaves inserted at various places.

Although the above comparison focuses only on version 1.0 of the Quattro Pro menu tree, the scope of copying remains the same for versions 2.0, 3.0, and 4.0. The only difference among the emulation menu trees in Quattro Pro versions 1.0, 2.0, 3.0, and 4.0, is that each successive version added new commands. *See* Warfield Dep'n, III:40 at 72-73. Thus, later versions also contain a virtually identical copy of the Lotus menu tree, with still more new branches or leaves added.

The presentation of the commands in the Quattro and Quattro Pro programs indicates which commands are found in the 1-2-3 menu tree and which are Borland's insertions. In Quattro, the new (and only the new) menu commands are followed by a dot at the place where the new command is inserted into the tree. Thus, in Quattro, the "Install" command was inserted into the first level menu followed by a dot. The submenu of "Install" does not have these same dot designations. Such a designation is unnecessary for this submenu, however, because everything within the newly inserted "Install" branch of the tree is new.

Quattro Pro uses the same approach for differentiating 1-2-3 menu commands and added menu commands, with the exception of "View" in Quattro Pro's first menu, which has no designation as an inserted command but whose submenu consists of commands each followed by a dot. In addition, in the Quattro Pro menus, a line generally appears separating added menu commands from original 1-2-3 menu commands.

The disparity in total number of menu commands between the 1-2-3 menu tree and the Quattro and Quattro Pro menu trees is large, but does not alter the scope of copying. Most of

the new commands are clustered together far up in the tree (or down in the upside down tree). For those menus that exist in the Lotus 1-2-3 menu tree, the number of added menu commands is rarely more than one or two. The effect is similar to an identical copy of a book with some paragraphs and lengthy footnotes inserted, and some voluminous appendices attached at the end.

In short, I adhere to my earlier conclusion that no reasonable factfinder could find that Borland did not take from Lotus 1-2-3 the menu commands and menu command structure substantially as they were. Further, I find that Borland produced a virtually identical copy of 1-2-3 menu tree, albeit with additions, in its Quattro and Quattro Pro emulation interfaces.

## **B. Nature of the Work—Scope of Illicit Copying.**

### **1. *Scope of limitations imposed by functional considerations.***

#### **a. *The functional considerations.***

Borland advanced the following "constraints" on design of a menu tree.

- (1) Each menu command was chosen to tell the user the purpose of the menu command and its function.
- (2) Each menu command was selected so that it had a different first letter from the other menu commands within the same menu.
- (3) Each menu was set up to have only about seven choices (witnesses referred to this as the "seven plus or minus two rule," *i.e.*, no menu should have fewer than five or more than nine commands).
- (4) The menus were structured so that words within the menu that dealt with similar functions were grouped together.

- (5) Executable operations that were likely to be frequently used were located near the top of the (upside down) tree.
- (6) Menu commands within a menu were arranged from left to right in order of decreasing frequency of use.
- (7) Commands in submenus were grouped under the menu command to which they relate.

*See VII:A:6.* Borland also proffered an eighth functional "constraint" of having each menu fit on one line of a computer screen. If this constraint were followed, each menu could have no more than 80 characters (the number of characters across a standard computer terminal screen).

Borland refers to the above items as "constraints" on design of a menu tree. Lotus contends that they are "guidelines" or "rules of thumb." The difference in terminology is largely if not entirely semantic. Nevertheless, the different terms raise different images. "Constraint" implies a rule that must not be violated. If "constraint" is defined this way, the eight listed items are not "constraints." Each of them is violated somewhere in the Lotus 1-2-3 menu tree (except possibly the second, which has been violated in other menu trees, *e.g.*, Excel 3.0 with "Save" and "Save As" in the same menu, and the last, which was violated in SuperCalc). Moreover, the alleged "constraints" are full of terms that offer little guidance on how to conform. For example, when is one menu command "similar" to another? The answer may be easy in some cases and entirely within the discretion of the programmer in others.

"Guideline," or "rule of thumb," is a more accurate description. Each of these terms implies a directive that is not always to be followed. Even these terms, however, ordinarily imply a precise rule. The first item listed above is hardly precise, even though it does limit the possible forms of a menu tree. Accordingly, I will refer to these as functional "considerations."

b. *Impact of functional considerations on form of the menu tree.*

The parties dispute the impact of the eight functional considerations (listed above) on the freedom of expression in forming a menu tree. For the following reasons I conclude that the Lotus 1-2-3 menu tree is just one of a great variety of possible expressions that are consistent with the functional considerations listed above and the specific set of executable operations used in Lotus 1-2-3.

First, none of the proffered functional considerations is overriding or dictates any specific result. Each may be violated. Most are violated at one or more points within the Lotus 1-2-3 menu tree. Indeed, the considerations are often competing and must be traded off against each other. *See, e.g.,* Gottheil Dep'n, IX:11 at 143-64; Olson Dep'n, III:32 at 117-18. No functional considerations are offered for guiding the determination of how to trade off competing concerns. Borland provided no credible evidence explaining how functional considerations could completely control formation of the menu tree.

Second, the "quality" of the menu tree depends on the peculiarities (*i.e.*, the particular tastes) of the individual using the program. *See, e.g.,* Bosworth Dep'n, I:1 at 389-90. For this to be true, there must be a variety of possible menu trees for the user to choose from. In this regard, Borland programs offer a "menu builder" that enables users to alter, customize, and create menu trees. If functional considerations restricted the possible expression of the menu tree to a limited number of possibilities, there would be little or no need for a user to modify it.

Third, even Borland's experts acknowledge that, given all of the various functional considerations, at least a limited range of choices remains for individual menu commands. *See* Liddle Dep'n, II:27 at 121; Olson Dec'n, V:13 at ¶ 35. Thus, "Copy" could be called "Replicate," "Duplicate," "Reproduce," "Repeat," "Ditto," etc. "Range" could be "Block," "Scope," "Extent," "Cells," etc. Although some experts contend that certain words and menu structure are preferable to

others, these same experts contend that the words Lotus selected did not matter for 1-2-3's success. *See* Liddle Dec'n, V:11, ¶ 55; Olson Dec'n, V:13, ¶¶ 36, 38. In any event, even if there were as few as two acceptable words for each menu command (given the other functional considerations), there would be 2 raised to the 469th power possible menu trees (an astronomical number) having precisely the same menu structure as the 1-2-3 menu tree, but with variations in menu command names. *See* Emery Dec'n, V:3, ¶ 85. Even if only half of the menu commands had more than one possible name (given the other functional considerations), there are over  $2^{234}$  possible menu trees having the same menu structure.

In listing these examples, I do not mean to suggest that the alternatives for menu command names are so few, nor to imply that every synonym for a command word is suitable in view of the other functional considerations. Rather, I merely note the breadth of possible menu trees that may be achieved in this manner as a factor bearing upon whether implementation of the idea, system, process, procedure, or method underlying the Lotus 1-2-3 menu tree is capable of a wide variety of expression.

Also, in presenting this analysis, I do not imply that counting possible variations in individual words is decisive of the breadth of possible expressions for literary works in general. It would not necessarily be significant, for example, to determine how many ways instructions for a simple game can be expressed solely by examining possible alternatives for individual words. Nevertheless, variations of the words in the menu tree represent a material consideration in the context of this case because of (1) the nature of the menu tree, which unlike an English sentence, permits substitution of individual words without changing the meaning of the menu tree, (2) the size of the menu tree, and (3) the structure of the menu tree.

If the designer of a menu tree chooses not to copy the structure of Lotus 1-2-3, the designer's freedom of expression and range of possible expressions for the menu tree expand dramatically. *See* Emery Dec'n, V:3, ¶ 84. Nothing in the materials before me supports an inference that functional considerations alone control the structure of the menu tree.

Indeed, Borland's experts implicitly acknowledge alternatives for the structure of the 1-2-3 menu tree. *See* Liddle Dec'n, V:11, ¶ 56; Olson Dec'n, V:13, ¶ 38.

Finally, a number of other spreadsheet programs use vastly different menu trees. The existence of vastly different menu trees in other commercial programs supports the conclusion that the Lotus 1-2-3 menu tree is but one of many possible forms for a menu tree.

Borland contends, however, that the differences in menu trees between programs such as Lotus 1-2-3 and Excel are due to differences in the programs' functionality. That is, Borland argues that because the executable operations (leaves of the tree) are different and the visual displays are different (*e.g.*, pop-up menus vs. text on a single line), the menu trees are different. Borland contends that, as a result, the existence of other commercial programs with menu trees vastly different from the 1-2-3 menu tree (and each other) is not probative of the degree of freedom of expression that exists in formulating the 1-2-3 menu tree.

I reject Borland's contentions for the following reasons. Although differences in program functionality may explain some differences between menu trees, the differences in functionality cannot explain the breadth of differences among menu trees used in the various programs. For example, functional considerations do not explain why the "File" menu command is left-most in the Excel menu tree and fifth from the left in the 1-2-3 menu tree. Borland does not suggest or offer any evidence explaining how the (unspecified) differences in functionality among 1-2-3, Excel, and Quattro's native mode affect placement of the "File" command within the first menu. Similarly, 1-2-3 places the copy and move command in the first level of the menu tree. In Excel they are in the second level.

The differences among menu trees in the various programs submitted are so large that they are, in a practical sense if not literally, incapable of enumeration. The broad scope of these differences cannot be explained in terms of differences in functionality. Indeed, Borland offers no evidence or argument providing a reasoned explanation (as opposed to an unsup-

ported assertion) of how the magnitude of differences could be explained by any differences in functionality. I conclude that many of the differences are due to different choices about how to express to the user the available user choices about all the particular operations that the program can perform.

In argument, Borland placed primary emphasis on an expert's assertion that the set of possible menu command names is "not large." *See* Olson Dep'n, III:32 at 54-59. As a preliminary matter, I find that this testimony is entitled to little, if any, weight. Nothing purports to explain how the expert reached this conclusion. In fact, this expert admits that she has never attempted to determine how large the set is. An expert's unexplained and unsupported assertion, even if received into evidence, is scant basis for reaching a reasoned finding.

In addition, the testimony is unclear whether the expert meant that there was a small set of choices for individual command names, or the set of all menu command names. In an earlier declaration this expert conceded that there is a "very narrow range" of suitable words for individual commands. *See* Olson Dec'n, V:13 at ¶¶ 35-36. As observed above, even if (given all the functional considerations) only half the commands have only one alternative label, the possible number of menu trees that differ in detail remains very high. Consequently, what this expert meant by "not a large set" may be entirely consistent with a determination that the number of possible ways of expressing the menu tree is without limitations that are material to the separation between idea or function and expression.

Finally, even assuming that the alternatives for menu command names were few, when I take into account as well the options available for structure, I find that the number of possible choices again expands dramatically. Thus, what Borland copied from 1-2-3 was not limited to aspects dictated by functional considerations. Rather, Borland copied the entire menu tree, much of which was the free expression of the creators of Lotus 1-2-3.

### c. Conclusion

As Borland practically conceded in closing argument, application of functional considerations does not restrict the expression of the menu tree to essentially only one form. (See Docket 333 at 3-52.) Further, I find that although functional considerations may have some effect on the design of a menu tree, they do not impose any practical limitation on the possible forms of expression to a number far enough short of infinity that any way of expressing the number in English words has come into common usage. The set of executable operations in Lotus 1-2-3 is large and the possible structural variations are enormous. The menu tree is capable of a very wide variety of expressions.

Borland has not argued or provided any evidence that any specific aspect of menu structure or command names, short of the entire menu tree, is dictated solely or influenced mainly by functional considerations. Moreover, Borland has used a virtually identical copy of the 1-2-3 menu tree. Accordingly, this case on its facts does not present any issue that might arise on a finding of copying of something short of virtually the entire menu tree.

### 2. Originality.

In its closing argument, Borland asserted that, because of functional considerations, creation of the menu tree did not require sufficient originality to justify protection under copyright law.

#### a. Originality doctrine.

The Supreme Court recently addressed the requirement of originality in *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991). The Opinion for the Court in *Feist* observed that the requirement that a work be "original" derives from the Constitution. The Court described the originality requirement as follows.

The *sine qua non* of copyright is originality. To qualify for copyright protection, a work must be original to the

author. Original, as the term is used in copyright, means only that the work was independently created by the author (as opposed to copied from other works), and that it possesses at least some minimal degree of creativity. To be sure, the requisite level of creativity is extremely low; even a slight amount will suffice. The vast majority of works make the grade quite easily, as they possess some creative spark, no matter how crude, humble or obvious it might be.

*Id.*, 499 U.S. at \_\_\_, 111 S.Ct. at 1287 (internal quotation marks and citations omitted).

The Court held that copying from the white pages of a telephone book was permissible. To reach this conclusion, the Court first confirmed the firmly established principle that facts (e.g., the names, towns and telephone numbers in the white pages) are not copyrightable. The Court also confirmed that "even a directory that contains absolutely no protectable written expression, only facts, meets the constitutional minimum for copyright protection if it features an original selection or arrangement." *Id.*, 499 U.S. at \_\_\_, 111 S.Ct. at 1289. In this circumstance, "the copyright is limited to the particular selection or arrangement." *Id.*, 499 U.S. at \_\_\_, 111 S.Ct. at 1290.

Applying these two basic principles, the Court determined that the selection and arrangement of facts in the plaintiff's telephone book, *i.e.*, selection of all people who applied for phone service and arrangement in alphabetical order, was "devoid of even the slightest trace of creativity." *Id.*, 499 U.S. at \_\_\_, 111 S.Ct. at 1296.

[O]riginality is not a stringent standard; it does not require that facts be presented in an innovative and surprising way. It is equally true, however, that the selection or arrangement of facts cannot be so mechanical or routine as to require no creativity whatsoever.

*Id.* Indeed, the selection of all people and placement in alphabetical order was not at all original to the plaintiff—phone books had been arranged in that fashion for years. Plaintiffs

white pages were nothing but a "garden-variety" phone book. *Id.*

*b. Originality in this case.*

Relying on *Feist*, Borland contends that the eight "rules" regarding functionality listed above demonstrate that the form of the menu tree (including menu command names and structure) is not original. For the following reason, I find that the menu tree is "original."

As explained above, the menu tree of Lotus 1-2-3 may be viewed as an arrangement of the executable operations as "leaves" of the "menu tree," with internal commands ("branches") leading through the structure to the "leaves." Altogether, the tree contains 469 labels to explain the arrangement. This text and arrangement presents the set of executable operations to the program user. Thus, the specific operations could be viewed as (uncopyrightable) facts and the menu tree as an arrangement (with textual labels) of the facts. Even viewed in this light, most favorably to Borland, the menu tree easily satisfies the originality requirement discussed in *Feist*.

The functional considerations Borland propounds are not at all comparable to application of a rule of alphabetical ordering of executable operations.

Borland's "rules" are violated with regularity in the 1-2-3 menu tree. In contrast, the rule of alphabetical ordering was not violated in the *Feist* white pages.

Also, the functional considerations listed above were not "an age old practice, firmly rooted in tradition and common place." *Feist*, 499 U.S. at \_\_\_, 111 S.Ct. at 1297.

Moreover (except in a few places where rules of alphabetical ordering differ) one who is given the names, addresses and numbers in a phone book, can generate the alphabetical ordering by routine and mechanical application of rules, with only one possible outcome. In contrast, the text for the menu commands and the menu structure itself are not dictated by mechanical application of the functional considerations. As noted above, a wide variety of menu trees is possible.

In sum, the 1-2-3 menu tree "make[s] the grade quite easily, as it possess[es] some creative spark. . . ." *Feist*, 499 U.S. \_\_\_, at 111 S.Ct. at 1287.

Borland nevertheless contends that, because functional considerations played a role in formulating the 1-2-3 menu tree, the menu tree is not copyrightable. Borland's argument is susceptible of two interpretations, both lacking merit.

First, one may interpret this argument as a contention that the functional considerations so permeate formulation of the menu tree that the menu tree is not separable from the "idea" of the program. This form of argument has nothing to do with the amount of creativity or originality involved. Derivation of a scientific formula may require a great deal of creativity and produce an original result. If the formula fails the copyrightability test, it is because the formula is not expressive—independently of creativity or originality. Casting the argument in terms of originality doctrine rather than separability does nothing to assist resolution of the issues in this case, and may lead to confusion. In any event, to the extent Borland raises the same separability argument that it has raised before, but dressed now in terms of originality, that argument was previously rejected by the court for good reason, and is now rejected again.

Alternatively, Borland may be understood as contending that any work whose form is restricted to any material extent by functional considerations is not original. Without more, this contention is invalid on its face. Any original literary work is formulated according to functional considerations imposed by language, a desire for clarity, and a desire to express the ideas conveyed. As one of Borland's experts concedes, the first functional consideration (conveying the nature of executable operations) is not materially different from a functional consideration for selection of words in any English writing. See Olson Dep'n, III:32 at 54. Accordingly, this type of "functional consideration" can remove a writing from copyright protection only if it restricts the forms of expression (that are separable from the idea or function of the work) to a limited number. Thus, cases referring to functional con-

siderations and decided under originality doctrine, including those cited by Borland, uniformly refer to limits on the number of forms of expression given functional considerations.

*See Victor Lalli Enter., Inc. v. Big Red Apple, Inc.*, 936 F.2d 671, 673 (2d Cir.1991) ("purely functional grids that offer no opportunity for variation"); *Sinai v. California Bureau of Automotive Repair*, 25 U.S.P.Q.2d 1809 [1992 WL 470699] (N.D.Cal.1992) ("limited number of ways" to arrange information in chart); *New Haven Copper Co. v. Eveready Mach. Co.*, 229 U.S.P.Q. 838 [1986 WL 269] (D.Conn.1986) (column headings "dictated by functional considerations"); *Merritt Forbes & Co. v. Newman Inv. Sec., Inc.*, 604 F.Supp. 943, 951-52 (S.D.N.Y.1985) (no originality if "form of expression is dictated solely by functional considerations"; determining that fact issue exists over whether concept is "capable of varied expressions"); *Decorative Aides Corp. v. Staple Sewing Aides Corp.*, 497 F.Supp. 154, 157 (S.D.N.Y.1980) (what was similar was "dictated by functional considerations"), *aff'd without op.*, 657 F.2d 262 (2d Cir.1981). *Cf. Feist*, [499 U.S. at \_\_\_], 111 S.Ct. at 1296 (routine and mechanical application of single rule).

Thus, Borland's argument fails because, as above, a wide variety of expression is possible for the Lotus 1-2-3 menu tree.

### C. Value of Expert Testimony.

In reaching the conclusions above, I have read and weighed the expert testimony offered by the parties. For the following reasons, however, much of this testimony has little weight in relation to the issues of law and fact that are decisive of the outcome in this case.

First, most of the testimony expresses conclusions, without any reasoned explanation of the basis for the conclusions. Second, much of the testimony uses terms such as "system" or

"a large number" that the experts do not define. As factfinder, I choose not to abandon my factfinding responsibility by accepting an expert's proposed unexplained choice. This is my usual practice, and for even stronger reasons I proceed in this way when the expert's testimony uses terms specially defined by the expert to have meanings that incorporate into the terminology substantive choices that have important policy implications, without examining and explaining reasons for the policy choices and whether they are consistent with applicable statutes and precedents.

Finally, much of the testimony is based on explicit or implicit assumptions about copyright law that are incorrect. For example, one expert testified that she was unsure of whether design of a menu tree is creative. In defining creativity, however, the expert explained that creativity required some "inventive leap." Olson Dec'n, III:32 at 124-35. Of course, this type of creativity—the inventive leap or new idea—is not required for copyrightable expression. *See Feist*, 499 U.S. at \_\_\_, 111 S.Ct. at 1296 (novelty or innovation not required). Expression is copyrightable, even when it is a new expression (*e.g.*, a new novel about young love) of an old idea (*i.e.*, boy meets girl).

### D. Conclusion.

In sum, I conclude that each of the Borland emulation interfaces contains a virtually identical copy of the 1-2-3 menu tree and that the 1-2-3 menu tree is capable of a wide variety of expression.

## III. Affirmative Defenses for Phase I.

### A. Laches.

Borland has the burden of proving laches. *Costello v. United States*, 365 U.S. 265, 282, 81 S.Ct. 534, 543, 5 L.Ed.2d 551 (1961). A defense of laches has been established if the defendant proves that (1) the plaintiff inexcusably or

unreasonably delayed in bringing an infringement action, and (2) defendant was prejudiced by this delay. *See id.* *See also* 3 Nimmer on Copyright § 12.06.

The chronology of the principal events bearing upon Borland's defense of laches commences in January 1987, when Lotus filed suit in the *Paperback* case. Borland was developing Quattro at the time. Before including the 1-2-3 emulation interface in Quattro, Borland secured a legal opinion that the program did not infringe Lotus's copyright. In late September 1987, Borland announced Quattro, Borland's first spreadsheet product. In November, Borland shipped the Quattro product. Quattro was advertised widely, one advertisement appearing in a November 1987 issue of "Lotus Magazine."

In early 1988, the Quattro product was discussed at a high-level Lotus meeting. The evidence concerning this meeting supports an inference that certain Lotus executives discussed the possibility that Borland would change Quattro if Lotus threatened suit. The evidence does not prove, however, that Lotus delayed filing suit or kept silent on the issue of infringement *for this reason*. I find that Lotus may have discussed the possibility that Borland would change its product, but that this was not a factor in Lotus's decision not to file suit in 1988.

Some time after the meeting, Lotus decided not to file suit against Borland then. In addition, Lotus adopted a policy of not commenting on possible infringement by, or legal action against, Borland.

In September 1988, less than one year after Borland shipped Quattro, Borland acquired the "Surpass" spreadsheet technology (from "Surpass Software Systems") for approximately \$2.4 million. This included a software program that had been called "Surpass." The Surpass program's only menu tree was a copy of the 1-2-3 menu tree. After acquiring Surpass, Borland removed the Surpass product from the market. Borland then re-engineered Surpass to become the Quattro Pro product, having a 1-2-3 compatible menu tree as just one of the possible menu trees available. Quattro Pro version 1.0 was released in November 1989.

Near the end of June 1990, this court handed down its decision in the *Paperback* case. The day after the decision was released, Borland filed an action in California seeking a declaration of noninfringement. On July 2, 1990, Lotus filed the original complaint in the present action.

### 1. Delay.

The period between Borland's release of Quattro and Lotus's filing of the instant action was approximately 2½ years. This amount of time, standing alone, does not dictate a finding that the delay was unreasonable.

*Cf. Roulo v. Russ Berrie & Co.*, 886 F.2d 931, 942 (7th Cir.1989) ("A two year delay in filing an action following knowledge of the infringement has rarely been held sufficient to constitute laches."), *cert. denied*, 493 U.S. 1075 [110 S.Ct. 1124, 107 L.Ed.2d 1030] (1990); *Hoste v. Radio Corp. of America*, 654 F.2d 11 (6th Cir.1981) (per curiam) (error to grant summary judgment based on laches for suit filed after 13 year delay); *Boothroyd Dewhurst, Inc. v. Poli*, 783 F.Supp. 670, 680 (D.Mass 1991) (four year delay may not constitute laches).

Whether this amount of delay was unreasonable or inexcusable depends on the motives for delay. Thus, Judge Learned Hand observed that

it is inequitable for the owner of a copyright, with full notice of an intended infringement, to stand inactive while the proposed infringer spends large sums of money in its exploitation, and to intervene only when his speculation has proved a success. Delay under such circumstances allows the owner to speculate without risk with the other's money; he cannot possibly lose, and he may win.

*Haas v. Feist*, 234 F. 105, 108 (S.D.N.Y. 1916). On the other hand, delay used to evaluate and prepare a complicated claim may be reasonable. *See, e.g., Paperback*, 740 F.Supp. at 82.

Similarly, delay to determine whether the scope of proposed infringement will justify the cost of litigation may be permissible. *See, e.g., Boothroyd Dewhurst*, 783 F.Supp. at 680-81. In the circumstances of the present case, I find that Lotus's delay in filing suit was reasonable and excusable.

First, I reject Borland's contention that Lotus's delay in bringing suit is of the kind addressed in *Haas*, i.e., in order to speculate with Borland's money. Borland relies primarily on the failure of Lotus to file an action in the early part of 1988. Borland contends that Lotus then knew enough about Borland's plans that Lotus should have notified Borland immediately if Lotus intended to claim infringement and that Lotus did not do so because Lotus was afraid that Borland would change Borland's product. Borland contends, in other words, that Lotus waited to file suit in an effort to trap Borland into expending large sums of money on Borland's spreadsheet products and, later, to reap the benefits of Borland's investment.

I find, however, that the evidence Borland presented does not support Borland's contentions. At the time of the 1988 meeting, Borland had already developed, advertised, and shipped Quattro. The possibility that Borland would adjust Quattro to remove the copy of the Lotus 1-2-3 menu tree was marginal in significance at the time of this meeting. For this reason, reinforced by the findings stated above, I do not find the evidence sufficient to persuade me, as factfinder, to draw the inference that Lotus waited to sue hoping that its competitor's product would succeed and enable Lotus to reap enhanced damages.

*See Russell v. Price*, 612 F.2d 1123, 1126 (9th Cir.1979) (conclusion that plaintiff was not speculating on defendant's success supported by fact that plaintiff was also seeking to exploit the copyrighted material), *cert. denied sub nom., Drebin v. Russell*, 446 U.S. 952 [100 S.Ct. 2919, 64 L.Ed.2d 809] (1980).

I find that Lotus delayed bringing suit for the purpose of awaiting resolution of the *Paperback* and *Mosaic* cases.

Throughout 1988, Lotus was involved in extensive litigation in *Paperback* and *Mosaic*. This litigation was hotly contested. If Lotus had lost the *Paperback* case, Lotus's claims against Borland would have been difficult to maintain. Thus, by waiting to file suit, Lotus avoided expensive duplicative litigation all of which might have been unsuccessful if Lotus lost in *Paperback*. This avoided a risk of needlessly wasting court and party resources. Once the *Paperback* case was resolved by the trial court, Lotus filed suit against Borland almost immediately. I cannot say that this was unreasonable.

Borland contends, nevertheless, that Lotus's delay was unreasonable in part because Lotus kept silent about its belief that Borland's product infringed. Lotus responds that it was satisfied that Borland knew of the infringement claims because of (1) Phillippe Kahn's (Borland's chairman's) strong public reaction condemning Lotus for filing the *Paperback* case, and (2) a public statement Lotus issued before Quattro was shipped that appeared in the May 26, 1987 issue of PC Magazine (Exh. 28 at 162-63). In the public statement, Lotus represented that it did not claim copyright over the ideas of a two-line moving cursor interface or context sensitive help. Lotus then characterized its suit against Paperback and Mosaic as protecting against companies that copied "all of the ways in which 1-2-3 communicates to the user, including its menu structure and sequence, word selection, and macro language design. . . ."

I find that Lotus reasonably believed that Borland was aware of Lotus's potential claims against Quattro and Quattro Pro. Borland "was far too sophisticated to need being led by the hand." *Famous Music Corp. v. Bay State Harness Horse Racing & Breeding Assoc., Inc.*, 554 F.2d 1213, 1215 (1st Cir.1977) (rejecting estoppel defense). In light of this, I cannot say that Lotus's decision not to comment on the infringing nature of Quattro or Quattro Pro makes Lotus's delay in this case unreasonable or inexcusable.

In weighing equities, some courts decline to permit a laches defense by a "deliberate pirate," *Haas*, 234 F. at 108, or less harshly stated, a "deliberate infringer."

See 3 Nimmer on Copyrights, § 12.06 at 12-107 (delay "may not be a bar against one who knew of plaintiffs asserted rights, or as against a deliberate infringer") (footnotes omitted).

I find that Borland (1) knew of Lotus's copyright, (2) knew that Lotus was acting to protect that copyright, and (3) deliberately introduced a virtually identical copy of the Lotus 1-2-3 menu tree into its 1-2-3 emulation interfaces.

The fact that Borland did secure a noninfringement opinion from a prestigious law firm is relevant, of course. Nevertheless, Lotus is not in any way accountable for that opinion. Indeed, Borland offers no evidence that Borland gave any notice to Lotus of Borland's seeking legal advice or of the substance of that advice. It is a rather curious twist of argument to suggest that a copyright owner has a duty of notice to the infringer when the infringer has enough concern about its own actions that it seeks a legal opinion privately but refrains from giving any kind of notice to the copyright owner.

Borland contends that the *Paperback* suit could not constitute notice that Lotus was enforcing its copyright because the issues in *Paperback* and this case are now identical. The fact that the issues are not identical, however, is far from decisive. More to the point is the fact that Borland was aware of Lotus's assertion of copyright protection for the 1-2-3 program and in particular for the form of its menu tree.

Borland next argues that Lotus had not sufficiently formulated its precise contentions of what aspects of Borland's products infringe Lotus's copyright until May of 1992—almost two years into this case. Again, this has little to do with the fact that Borland was aware of Lotus's assertion of copyright protection for Lotus 1-2-3 and in particular for the form of its menu tree.

## 2. Prejudice.

The defense of laches requires that defendant suffer prejudice. Typical forms of prejudice include: death or unavailability of an important witness, dulling of memories, loss of relevant records, and continuing investments and outlays by

the alleged infringer in connection with the operation of its business. See *Eisenman Chemical Co. v. NL Indus., Inc.*, 595 F.Supp. 141, 147 (D.Nev.1984). Borland argues only the last form of prejudice. To demonstrate prejudice, Borland points to marketing costs and acquisition of the Surpass technology.

To constitute prejudice, however, these expenses must have been incurred *as a result* of Lotus's delay in bringing suit. Where an infringer was aware of a plaintiff's copyright, as Borland indisputably was, courts have phrased this point as a requirement that the actions be taken in reliance on the plaintiff's delay in bringing suit.

See *Russell*, 612 F.2d at 1126; *In Design v. Lauren Knitwear Corp.*, 782 F.Supp. 824, 831 (S.D.N.Y.1991); *Steinberg v. Columbia Pictures Indus., Inc.*, 663 F.Supp. 706, 716 (S.D.N.Y.1987).

For the following reasons, I conclude that Borland has not proved that it took any action in reliance on Lotus's delay in bringing suit.

First, I find that Borland would have invested in and entered the spreadsheet market whether or not Lotus delayed in bringing suit. Phillippe Kahn testified that the decision to include the 1-2-3 emulation interface in Quattro was made not long before the product shipped. (I:14 at 362). Of course, this initial decision to include the 1-2-3 emulation interface was made before Lotus could have objected.

The Surpass technology, acquired in 1988, was used to form the basis for Quattro Pro. This was less than one year after Borland shipped Quattro. Borland presented no evidence that it would not have acquired Surpass had Lotus taken action earlier. Moreover, Quattro already had a 1-2-3 emulation interface. It would not be reasonable to infer that other aspects of Surpass than its copy of the Lotus 1-2-3 menu tree had no influence on Borland's decision. I find that Borland has not proved that it acquired the Surpass technology in reliance on Lotus's delay in bringing suit.

I observe also that Borland filed a declaratory judgment action immediately after the decision in the *Paperback* case.

Borland seeks to explain the timing of the filing of the declaratory judgment action by presenting evidence that Borland was responding to recent conflicting reports that Lotus would file suit against Borland. Even if I credited this self-serving assertion, unsupported by any objective evidence, however, I would not infer that Borland had ever relied on Lotus's failure to assert infringement to conclude that Lotus would never file suit. Moreover, a senior Borland official averred that the Borland Board of Directors had been informed before the decision in the *Paperback* case that Lotus intended to sue Borland if Lotus prevailed in *Paperback*. (Leyton Dec'n, VI:10, ¶ 5.) Nevertheless Borland, like Lotus, awaited the outcome of the *Paperback* case. Significantly, when Borland filed, it sought a declaration of noninfringement, but omitted any specific request for declaration of unenforceability due to laches or estoppel.

Based on the evidence before me, I find that Borland developed and marketed Quattro and Quattro Pro in reliance on (1) the noninfringement opinion, and (2) the hope that Lotus would lose the *Paperback* litigation. Further, I find that Borland has not proved that it took any actions in reliance on Lotus's delay in bringing suit. Finally, I find that reliance, even if it had occurred, would have been unreasonable in the circumstances shown by the evidence in this case.

### B. Estoppel.

The parties apparently agree that, to establish a defense of estoppel, Borland must prove that Lotus engaged in (1) conduct that induced Borland to change its position in good faith, or (2) conduct on which a reasonable person would rely. See *Precious Metals Associates, Inc. v. Commodity Futures Trading Comm'n*, 620 F.2d 900, 908 (1st Cir.1980). In view of this agreement, I need not and do not consider whether this interpretation of *Precious Metals* and other precedents is more generous to Borland's estoppel claim than a more precise understanding of the precedents would be.

To establish estoppel, Borland relies primarily on Lotus's delay in bringing suit and Lotus's failure to inform Borland of Lotus's belief that the Borland products infringe. For the reasons explained in Section A above, I find that (1) Lotus did not intend that Borland infer from Lotus's silence that copying of the menu tree was permissible, (2) Borland was aware of Lotus's copyright, (3) Borland did not rely on the silence of Lotus to Borland's detriment, and (4) any reliance by Borland in the circumstances of this case would have been unreasonable and unjustifiable.

Borland also offers a collection of actions or statements by Lotus that Borland contends entitled Borland to infer that Lotus would not sue Borland for copyright infringement. Borland cites a January 1987 InfoWorld article, released before Lotus had any reason to know of Borland's Quattro product. The article purports to quote a Lotus spokesman, Greg Jarboe:

"Some folks have misinterpreted that what we have done is to copyright the spreadsheet or the two-line [command] interface, neither of which we are trying to do," Jarboe said. The suit was targeted at two vendors who had "copied 99 percent" of a Lotus product, he said, adding that the company is considering issuing a position statement to clarify the limits of its copyright.

InfoWorld, January 26, 1987. Even assuming this is an accurate quote, something Mr. Jarboe denies, I find that it would not be reasonable to launch a spreadsheet product in reliance on this statement. First, as Borland's general counsel concedes, II:20 at 272, the statement does not purport to disavow filing suit against future products that infringe Lotus's copyright in a manner different from the *Paperback* suit. A characterization of the *Paperback* case as involving "clones" is not reasonably interpreted as expressing an intention not to sue other infringers as well as makers of "clones." Second, the article refers to an upcoming position statement (the PC Magazine statement quoted in Section III.A, above). Although Borland contends that none of Borland's responsible representatives saw this position statement (a contention that I find

questionable in light of their other activities), their failure to see and take account of that position statement, along with acting on the assumption that no such statement had been made, was unreasonable. Thus, to the extent Borland claims that it relied on the InfoWorld statement, it was unreasonable for Borland to rely on that statement while failing to take further steps to determine what Lotus's position statement contained.

Borland next cites the fact that Lotus Magazine published advertisements for Borland's products. I find that it would not be reasonable for Borland to assume that the acceptance of the advertisement in Lotus Magazine showed that Lotus had determined that it would make no infringement claim against Borland. Following Borland's line of reasoning, one would conclude that by allowing competitors to run advertisements in Lotus Magazine, Lotus Development Corporation agreed that the competitor's advertising claims were correct (*e.g.*, that Borland's products are superior, as the advertisements implied) and that Lotus wanted individuals to purchase from Lotus's competitors. Merely stating this chain of inference is enough to expose its unreasonableness. I find, instead, that the fact that Lotus Magazine accepted the Borland advertisement indicated a policy of accepting advertisements—even from competitors. Moreover, on the evidence before me I cannot find, as factfinder, that the plaintiff in this case, Lotus Development Corporation, controlled the decision-making process about what appears in Lotus Magazine. Accordingly, it was not reasonable for Borland to infer that Lotus Development Corporation was making any statement concerning possible copyright infringement when Lotus Magazine accepted the advertisement.

Borland next cites examples of occasions where Lotus personnel interacted with Borland personnel at trade shows and with respect to new products and services, without mentioning any belief that the Borland products infringe. Nothing in these materials suggests that the Lotus personnel involved had the actual or apparent authority, or even sufficient knowledge, to comment on whether or not Lotus believed that Borland

infringed a copyright or whether Lotus intended to file an action in the future. Moreover, nothing in the record shows that the individuals involved actually said anything that would support an inference by Borland that copying of the menu tree was permissible. Accordingly, I find that Borland could not reasonably interpret these communications to mean that Lotus would not file suit.

In sum, I find that both separately and taken as a whole, the events, conduct, and communications that Borland points to in an effort to bolster claims for estoppel fall far short of supporting estoppel.

### C. Conclusion.

For the reasons explained above, I find that Lotus's claims in Phase I of the trial are not barred by laches or estoppel. In presenting its case on the affirmative defenses, Borland offers a number of proposed inferences that this court might draw, based loosely on evidence before the court. Although I have not explicitly referred to each of Borland's individual contentions in this Opinion, I have examined all of the contentions and have found them without merit.

UNITED STATES DISTRICT COURT  
D. MASSACHUSETTS

Civ. A. No. 90-11662-K

July 31, 1992

LOTUS DEVELOPMENT CORPORATION,

*Plaintiff,*

—v.—

BORLAND INTERNATIONAL, INC.,

*Defendant.*

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, Mass., Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, Mass., Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, Cal., David L. Hayes, Mitchell Zimmerman, Fenwick & West, Palo Alto, Cal., Peter Erich Gelhaar, Donnelly, Conroy & Gelhaar, Boston, Mass., Gary L. Reback, Wilson Sonsini, Goodrich & Rosati, Palo Alto, Cal., *for defendant.*

MEMORANDUM AND ORDER

KEETON, *District Judge.*

By Memorandum and Order of March 20, 1992, the court dismissed the parties' motions for summary judgment in this copyright infringement action and invited new motions compatible with rulings therein announced. Each party has renewed its motion for summary judgment and filed further

submissions (Docket Nos. 168-190).<sup>\*</sup> A hearing on these motions was held on May 19, 1992, and additional submissions were filed after that hearing.

The reader may find background information in two earlier documents issued by this court: the first, the Opinion in a related case involving the Lotus 1-2-3 program at issue here, *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass. 1990), and, the second, a Memorandum issued in this case, *Lotus Dev. Corp. v. Borland Int'l Inc.*, 788 F.Supp. 78 (D.Mass. 1992) (Memorandum and Order of March 20).

In the Memorandum and Order of March 20, I concluded that Lotus had failed to frame adequately its contentions with respect to the infringement of elements of its user interface less than the whole interface. In its renewed motion for summary judgment, Lotus asserts specifically that Borland has copied expressive elements of the 1-2-3 interface, including "menu commands," "menu structure," "long prompts," and "keystroke sequences." Although Lotus continues to argue that its entire user interface beyond dispute was copied, I adhere to the view that on the present record a reasonable jury could find that Borland copied less than the whole 1-2-3 user interface. Nevertheless, based upon the parties' most recent submissions, I conclude that, beyond *genuine* dispute, Fed.R.Civ.P. 56, Borland copied parts of the 1-2-3 user interface. For the reasons stated herein, I deny Borland's motion for summary judgment and grant, in part, Lotus' motion for summary judgment.

<sup>\*</sup> Lotus argues that Borland has not filed a renewed motion for summary judgment. There is no docket notation of Borland's renewed motion, nor has such a motion been physically located by the court. However, several of Borland's submissions are submitted in support of "Borland's Renewed Motion for Summary Judgment." In this circumstance, and because I conclude that I must in any event deny such a motion, I will consider Borland's "motion" on its merits.

## I. DEFINITIONS AND PREMISES

The Memorandum of March 20 presented for possible use in this case the following form of jury interrogatory concerning the extent to which Borland copied the Lotus 1-2-3 user interface in creating its Quattro programs:

## Question 1

- (a) Do you find that the Quattro Pro user interface as a whole was copied from the Lotus 1-2-3 user interface as a whole?

\_\_\_YES \_\_\_NO

- (b) Do you find that the part of the Quattro Pro user interface called the "emulation interface" (also called the "1-2-3 compatible interface") as a whole was copied from the Lotus 1-2-3 user interface as a whole?

\_\_\_YES \_\_\_NO

- (c) If NO, do you find that some part, and, if so, which of the following part or parts of the Lotus 1-2-3 user interface were copied into some part of the Quattro Pro "emulation interface" (also called the "1-2-3 compatible interface")?

- |                          |        |       |
|--------------------------|--------|-------|
| (1) The menu commands    | ___YES | ___NO |
| (2) The menu structure   | ___YES | ___NO |
| (3) The command sequence | ___YES | ___NO |
| (4) The long prompts     | ___YES | ___NO |
| (5) The macro facility   | ___YES | ___NO |

The Memorandum of March 20 noted that it was not clear that Lotus was making a claim of the sort addressed in part (a) of proposed Question 1 and that ambiguity remained regarding the meaning of the terms used in parts (c)(1)-(5). Lotus has in its recent submissions clarified its contentions.

First, Lotus acknowledges that the "native" modes of the Quattro programs have user interfaces that differ from that of 1-2-3. Thus, the question posed in part (a) of Question 1 is not in genuine dispute.

Second, Lotus has defined, as it uses them, the terms "menu commands," "menu structure," "keystrokes," "keystroke sequences," "long prompts," and "macro language."

In general, except for some blending of argument with definitions, the parties appear not to be in dispute about the meaning of these and other terms defined below. The definitions that I use in this Memorandum, for the purpose of explaining and analyzing the contentions of the parties, are consistent with the submissions of both Lotus and Borland, as I understand them.

"Command" refers to an abbreviated description of a direction that a user of a software program (whether Lotus 1-2-3, Borland's Quattro Pro, or another program) may invoke to cause some operation to be performed.

"Menu" refers to a display on the computer monitor of a limited number of commands available to the user at a given moment.

"Menu command" refers to a command that appears in a menu. In Lotus 1-2-3, a menu command is ordinarily a single English-language word. In rare instances, it is instead a representation of an English-language pronunciation (such as "Xtract"). Menu commands are displayed on the computer monitor by the 1-2-3 program in a succession of menus. The menus communicate to the user, in sequence, the spreadsheet operations available to the user.

"Command structure" refers to the organization of the menus and menu commands. (Other phrases used with essentially the same meaning include "menu command structure," "menu hierarchy," and "menu command hierarchy.") In Lotus 1-2-3, menu commands are organized so that less than a dozen related menu commands are displayed at any given moment. This display communicates to the user the spreadsheet operations immediately available. Each menu of less than a dozen commands is linked to preceding/succeeding menus by the operation of menu commands. All command menus are ultimately linked to a single main (root/trunk) menu to form a "menu tree."

"Keystroke sequence" refers to a sequence of keystroke entries that a user may invoke. Keystroke sequences may be

generated as one navigates the menu command hierarchy performing sequential spreadsheet operations.

"Long prompt" refers to a displayed multi-word English-language description of a "highlighted" menu command. A "highlighted" menu command appears on the computer monitor as a block of inverse video—that is, on a monochrome monitor with a black background on which characters are lit, a highlighted word appears as black letters within a lit block.

"Macro language" refers to a feature by which a user may define a very short keystroke sequence as equivalent to a longer keystroke sequence. Thus, a user may invoke the short keystroke sequence (a "macro") as a substitute for the longer keystroke sequence. In stating this definition, I omit a sophisticated programming capability available in 1-2-3 through its macro language feature that Lotus, as I understand its submissions, does not contend is involved in its claim of infringement in this action.

Having stated the definitions of the components of the user interface that I will use in this Memorandum, I now state additional points that I conclude are not in dispute about the relations among these definitions and associated matters.

The keystroke sequences and macro language have functionality. Typing ("inputting," in jargon) the first character of a command word invokes the command and causes the operation associated with the command word to be performed. (In many instances, a submenu associated with the command word is displayed.) The menu command hierarchy is a fundamental part of the functionality of keystroke sequences and the macro language. For example, the keystroke sequence `/RFC` directs the computer to format a range of numbers to appear as currency values because the character `/` initiates a command sequence, the character `R` implements the "Range" command, the character `F` implements the "Format" command, and the character `C` implements the "Currency" command.

It may be necessary to enter additional information to invoke a spreadsheet operation fully. For example, in order to implement an operation formatting a range of numbers as currency values, it is necessary to delimit the range of spread-

sheet cells to be formatted. In addition, it is necessary to specify the number of decimal places to be displayed. Variables such as range of cells and number of decimal places, for which values must be input each time an operation is to be performed, are called "parameters."

The authors of Lotus 1-2-3 made certain predictions about the value of each of the relevant parameters likely to be input for use with certain spreadsheet operations. Those predictions have been incorporated into Lotus 1-2-3 as suggestions; a user failing to specify a value for a parameter where it is necessary to supply one accepts the suggestion of 1-2-3's authors by default. The `/RFC` command set, for example, has associated with it two "default parameters." The default for range is "current cell," and the default for number of decimal places is two (i.e., dollars and cents). The user who prefers a different format (for example, whole dollars rather than dollars and cents) may enter a different number, zero rather than two (the default) to so indicate. In similar fashion, the user may supply a range different than the default value.

The menu structure will not permit the command "Currency" to be executed without first proceeding through the "Range" and "Format" commands. Indeed, inputting a `C` at a different point in the menu structure may cause a different command, such as "Copy," to execute.

The foregoing description identifies one way in which a menu command may be invoked—that is, by pressing the letter key corresponding to the English-language name of the command. A second way of invoking a menu command is to make use of the highlighting around a menu command. The user may use arrow keys on the keyboard to move the block of highlighting ("cursor") to an adjacent menu command. Depressing the "Enter" key (or "Return" or "`<—`" key, depending on the keyboard) invokes a highlighted command.

Because the macro language plays such a central role in the parties' contentions, and because it is an extraordinarily sophisticated element of Lotus 1-2-3, I recite some further examples of the use of the macro language.

A user may define a keystroke sequence with a macro by inputting the keystroke sequence in a spreadsheet cell and

assigning a macro label to that cell. For example, if a user enters the sequence "Hello~" (the tilde, "~," stands in for the "Enter" or "Return" or "<—i" key) in spreadsheet cell A1 and assigns the macro label "\H" to cell A1, then the user may cause the word "Hello" to appear in any other cell by invoking the abbreviated keystroke sequence "\H" instead of the longer sequence "H," "e," "l," "l," "o," "Enter." (The reverse slash, "\," signifies the "Alt" key, pressed and held in place while another key is pressed. Thus, the keystroke sequence "\H" consists of pressing the "h" key while the "Alt" key is simultaneously depressed.)

The capability of the program to enter, for example, the word "Hello" in any spreadsheet cell when the "\H" keystroke sequence is entered is functional. It is not protectable by copyright, 17 U.S.C. § 102(b), and it is not the subject of this case.

At a more sophisticated level, a user may construct a macro that implements menu commands. For example, a user may input the keystroke sequence "/RFC" in cell A1 and attach a macro label such as "\C" to that cell. This defines "\C" in the macro language subject to later redefinition—as "/RFC~." Invoking the "\C" keystroke sequence in any cell will cause the spreadsheet to format that cell to display numeric values as currency values. The ability of the computer to format the cell for currency values is functional. It is not copyrightable, *id.*, and it is not the subject of this case. The spreadsheet program is instructed to format the cell by the keystroke sequence "/", "R," "F," "C," "Enter," "Enter." That sequence invokes the menu commands "Range," "Format," and "Currency" (and accepts, by operation of the "Enter" key twice, two default parameters). Thus, the menu commands are an important part of the functionality of the macros. Lotus contends that the menu commands and the command structure are copyrightable expressive elements of the 1-2-3 user interface and that they are copyrightable expressive aspects of the macros. Borland contends that the macros, in their entirety, are an uncopyrightable system. These competing contentions, as well as others, are addressed below.

## II. COPYING

In Borland's Mem. in Supp. of Renewed Mot. for Summ. J. (Docket No. 168), Borland states the following:

Furthermore, it is undisputed in the record that Borland did not copy the 1-2-3 menu command hierarchy directly from any 1-2-3 version, including Release 2.01. Rather Borland employees reviewed books about 1-2-3, Release 2.01, written by third-parties, which books contain schematic or menu-tree type representations of the 1-2-3 menu command hierarchy. Borland used these third-party menu trees to construct 123-compatible menu hierarchies in their own products.

But Borland employees did not copy those menu trees, even from the third-party books. Rather, they viewed the menu trees and implemented into their own products the relationship of functions depicted in those menu trees.

*Id.* at 15 (footnote omitted). Borland contends that, on these facts, indisputably Borland did not "copy." Borland's contention, however, is based on its idiosyncratic use of the word "copy," and is fundamentally wrong. Instead, Borland's admissions establish beyond dispute that Borland did copy the menu commands and command structure of Lotus 1-2-3.

Borland argues that the menu command hierarchy is a "set of functional relationships" that is nowhere displayed in the 1-2-3 user interface. Thus, Borland argues, it did not and could not have "copied" the menu command structure. That argument simply fails. It is an argument about a fact—copying or not. Its premise bears instead not on the fact-copying or not—but upon the legal issue of copyrightability. That is, the premise of Borland's argument is that the menu command structure must be fixed in a tangible medium to be copyrightable. See 17 U.S.C. §§ 101-102. The admitted fact that the Quattro programs duplicate the set of "functional relationships" of Lotus 1-2-3 and were designed to do so is conclusive against Borland on the issue of copying that set of functional relationships. Thus, Borland has admitted that it

intentionally incorporated into its user interface the 1-2-3 menu commands and menu command hierarchy.

Moreover, I reject Borland's tangible-medium argument as applied to this case. The argument would be relevant, if at all, to copyrightability rather than copying, but because Borland has made the argument as if it had a bearing on copying, I will digress briefly to address it here. To be the subject of copyright protection, an expression must be fixed in a tangible medium. *Id.* § 102. All that is required in this regard is that the expression be embodied in a copy "by or under the authority of the author" in a form "sufficiently stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration." *Id.* § 101. The output of a computer program, at least insofar as it is typical of the program, predictable from it, and directed by the operation of the program, satisfies these requirements. The menu command hierarchy is part of the 1-2-3 program's output, is directed by the program, is identical each time the program runs, and may be perceived (and, as in this case, duplicated). It is irrelevant that the hierarchy cannot be perceived in its entirety at one moment (for example, in one screen display), just as it is irrelevant that the plot of a novel cannot be perceived from viewing one page. The menu command hierarchy is copyrightable subject matter.

Borland's argument that it copied from third-party sources rather than Lotus is equally without merit. Borland argues this point on the strength of the court's suggestion that Borland might prove that it did not copy Lotus' interface by demonstrating that instead it copied someone else's. If Borland had copied a third party's independently created menu command hierarchy that "happened" to duplicate Lotus' interface, Borland might be excused (at least from liability to Lotus, though not perhaps to the third party). Here, however, Borland has admitted copying from sources that, with or without permission from Lotus, copied from Lotus. Borland was aware that those sources copied from Lotus—Borland admits verifying the accuracy of the menu structure it generated from third-party sources by comparing the Quattro programs to 1-2-3. The fact that Borland used third-party sources as a means of

copying the Lotus 1-2-3 menu command hierarchy in no way excuses Borland's deliberate imitation of the Lotus menu structure.

Based on the foregoing, I conclude that, beyond genuine dispute, Borland copied the menu commands and menu command structure of Lotus 1-2-3. Moreover, Borland admits to copying the functionality of the keystroke sequences and macro language. By its own assertions, Borland's reason for copying the menu command structure was to obtain the benefit of its functionality. It follows that in fact Borland has admitted copying aspects of the keystroke command sequences and macro language that Lotus contends are expressive and copyrightable, although Borland chooses to describe what it did in another way.

Lotus has identified at this stage one additional element of its user interface that it claims Borland copied: the long prompts. It is clear from the record that the long prompts appearing in the Quattro programs differ in many instances from the long prompts in 1-2-3. On the other hand, in many other instances there is little or no difference. There is evidence that a Borland employee wrote the Quattro long prompts and did not "copy" from 1-2-3; however, she admits to looking at 1-2-3 to ensure that she correctly understood the Lotus commands. A reasonable jury could find on the basis of this evidence either way—that Borland did or that Borland did not copy the long prompts of 1-2-3. Therefore, I conclude that whether the long prompts were copied is a question for the jury. Moreover, because of this conclusion, it is apparent that a reasonable jury could find that Borland did not copy the 1-2-3 interface as a whole.

### III. COPYRIGHTABILITY

Having concluded that Borland copied the menu commands and menu command hierarchy as well as the keystroke sequences and macro language, I now proceed to determine whether those aspects of the 1-2-3 user interface, taken together, are copyrightable.

## A. Potential Fact Questions

I stated earlier my tentative conclusion that the application of the copyrightability standard is for the court and not a jury. *Borland*, 788 F.Supp. at 96. I invited counsel to respond to that tentative conclusion.

Lotus has fully endorsed the conclusion that copyrightability issues, at least in this case, are for the court. *Borland*, however, contends (in the alternative to *Borland's* own motion for summary judgment) that questions of copyrightability are for a jury. In response to the court's invitation, *Borland* proposes the following questions to be answered by a jury:

1. Does the Lotus 1-2-3 menu command hierarchy comprise a system, procedure or method of operation?
2. Is the 1-2-3 menu command hierarchy designed and used as a system for performing tasks using a spreadsheet program?
3. Are the command words of 1-2-3 and their order an inseparable part of a system for performing spreadsheet tasks?
4. Does the 1-2-3 menu command hierarchy enable a person to map out and execute a procedure for performing a particular spreadsheet task?
5. Is the 1-2-3 menu command hierarchy a means for issuing commands to the computer program to perform spreadsheet tasks?
6. Does the 1-2-3 menu command hierarchy provide a procedure or method of operating a spreadsheet program?
7. Was the 1-2-3 menu command hierarchy designed and arranged using functional rules or principles?
8. Was the 1-2-3 menu command hierarchy designed and arranged to maximize its efficiency and usefulness?
9. Is the 1-2-3 menu command hierarchy fundamental to a user's ability to execute macros written using 1-2-3?

10. Do the command words of 1-2-3 convey information to the user other than the choices of functions that are available?

11. Does the 1-2-3 menu command hierarchy explain to the user how to use the 1-2-3 program to perform spreadsheet tasks?

Liability Questions for the Trier of Fact (Docket No. 95) at 2-5. Under my rulings, it is irrelevant that the 1-2-3 interface includes functional elements or "comprises a system" so long as it also includes separable expressive elements. Thus, five of *Borland's* first six questions are irrelevant. Question 3, though relevant, is a mixed law-fact question that will not be asked of a jury for reasons previously stated. *Borland*, 788 F.Supp. at 94-96.

The seventh and eighth questions are more problematic. Those questions may have a bearing on whether the expressive elements of 1-2-3 are in fact separable from the functional aspects of the interface. As stated in the Memorandum of March 20, "[i]f the menu commands or menu command structure were dictated solely by functional concerns, then those elements may not be copyrightable." *Borland*, 788 F.Supp. at 97 (citing *Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1145 (2d Cir.1987)). Lotus argues that *Brandir* is not "strictly applicable" because computer programs are classified as "literary works" and not "pictorial, graphic, and sculptural works" by the Copyright Act. The law draws heavily on analogy, however, and computer programs, whatever their formal classification, like pictorial, graphic, and sculptural works, are useful articles. Moreover, though insisting that the burden on this question falls on *Borland*, Lotus acknowledges that any elements of its program that were functionally dictated are not copyrightable. Pl.'s Mem. of L. in Supp. of Renewed Mot. for Summ. J. on Infringement (Docket No. 171) at 34-35 n. 47.

Nevertheless, I conclude that no reasonable jury could find that the menu command hierarchy was limited to one or even several alternate designs at the time it was created. On the

basis of the evidence before me, a factfinder could conclude that some—but only some—subelements of the menu command hierarchy were functionally dictated. For these reasons, it is my tentative conclusion that there may be genuine fact disputes regarding the subject matter of Borland's seventh and eighth questions. Questions seven and eight, as formulated, however, are "evidentiary" rather than decisive or "ultimate" issues appropriate for use on a verdict form.

It remains to be determined whether an "ultimate question" on the subject matter of the seventh and eighth of Borland's suggestions can be formulated as an adjudicative fact question of the type that is normally the province of a jury and does not pose the substantial risks described in my earlier Memorandum, *see Borland*, 788 F.Supp. at 94-96.

Question nine assumes a premise that is fundamentally in error—an error that has been systemic in Borland's arguments throughout the course of this litigation. The problem, which may be described as a "chicken and egg" problem, is addressed in greater detail in Section C, below. At this point, I simply state my view that question nine should not be asked of a jury.

The answers to the last two questions are irrelevant. A "no" answer to question ten does not preclude a determination that the command words are expressive even though the expression may be limited—i.e., the words communicate the functions to which they are assigned. Similarly, the answer to question eleven is immaterial.

### B. Legal Issues on Copyrightability

In the Memorandum and Order of March 20, 1992, I concluded that, "absent further guidance from higher authority before the date of trial," *Borland*, 788 F.Supp. at 89, I would apply the following standard for deciding copyrightability issues:

FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives

that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation, some conception of the "idea," "system," "process," "procedure," or "method"—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

\* \* \*

SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work."

*Id.* at 90 (quoting *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37, 60-61 (D.Mass.1990)) (all emphasis omitted).

The Second Circuit has recently issued an opinion that bears significantly on issues of copyrightability and substantial similarity. In *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 1992 WL 139364, No. 91-7893, 1992 U.S.App. LEXIS 14305 (2d Cir. June 22 1992), the court announced an "Abstraction-Filtration-Comparison" test for determining "substantial similarity." The first two steps, "abstraction" and "filtration," are designed to define the idea of the program and to eliminate it (as well as other noncopyrightable subject matter) from further consideration. The test "serves 'the purpose of defining the scope of plaintiff's copyright.'" *Id.* (quoting *Brown Bag Software v. Symantec Corp.*, 960 F.2d

1465, 1476 (9th Cir.1992)). Thus, the first two steps of the Second Circuit's "substantial similarity" test concern what other courts and commentators have called "copyrightability." Only the third step, "comparison," addresses similarities between the copyrighted work and the allegedly infringing work.

To what extent does the Second Circuit's "Abstraction-Filtration-Comparison" test differ, either substantively or in methodology, from the combination of copyrightability and substantial similarity tests tentatively adopted for this case in my Memorandum and Order of March 20?

I conclude that the standard for determining copyrightability stated in my Memorandum and Order is compatible with the abstraction-filtration portion of the Second Circuit's test. The Second Circuit founded its abstraction step on the opinions of Judge Learned Hand that were also the foundation of the first step of the copyrightability test stated in my Memorandum and Order. The second step of that copyrightability test parallels the Second Circuit's "filtration" step.

The third step of the Second Circuit test, "comparison," serves two functions. The first concerns the issue addressed in the third step of the "copyrightability" test I have tentatively adopted for this case— whether the expressive elements of the allegedly copyrightable work are a substantial part of it. I conclude that in this respect the two tests are compatible substantively though different in methodology. The other function that the Second Circuit's "comparison" step serves is emphasized in the term used to identify it—"comparison." The comparison is between the relevant portions of the allegedly infringing work and the expressive elements of the allegedly copyrightable work to ascertain whether any part of the allegedly infringing work is similar to expressive elements of the allegedly copyrightable work that are a substantial part of the allegedly copyrightable work (i.e., whether there is substantial similarity in the mixed law-fact sense). I conclude, again, that in relation to this comparison, the Second Circuit's test and the combination of the "copyrightability" and "substantial similarity" tests I have adopted tentatively are compatible substantively, though different in methodology.

Borland argues that the decision and reasoning in *Computer Associates* are contrary to the *Paperback* decision, and, as well, to my Memorandum and Order of March 20. In one respect only, however, did the Second Circuit explicitly so indicate. It criticized an incentive-based reason stated at one point in the *Paperback* opinion. The criticized argument, however, was by no means essential to the outcome in *Paperback*, and acceptance or rejection of that argument is not likely to affect the outcome in this case. In other respects, my reasoning in *Paperback* and in my Memorandum and Order of March 20, 1992 in this case was substantively consistent with the opinion of *Computer Associates*, as I understand that opinion, though, as I have explained above, different in methodology.

A particular example of significance to this case concerns Borland's argument that the Second Circuit's treatment of "compatibility" militates against copyrightability of the 1-2-3 interface. Borland extracts from the Second Circuit's opinion a determination that aspects of computer software cannot be subject to copyright if they are greatly circumscribed by the hardware or software with which they are designed to interact. That proposition, even if accepted as Borland has stated it, does not apply to 1-2-3. Borland's argument to the contrary must be rejected for reasons that I explain in Part C, immediately below.

### C. Which Came First?

A familiar childhood riddle asks: Which came first—the chicken or the egg? As folk riddles often do (and lawyer's questions on cross-examination sometimes do), this riddle strongly suggests that any answer but one of two identified options is out of bounds. Perhaps from another perspective, however, one may recognize that the chicken and egg are one type of organism in different stages of a life cycle. Over a long span of time the form of that life cycle has evolved from an earlier form in which the chicken and the egg were not so distinct as they now appear to be. History, science, and

philosophy may provide another answer, or many others, different from the two suggested in the riddle.

Borland's brief poses a conundrum for this case in a form analogous to the chicken-and-egg riddle. Borland asserts that the 1-2-3 interface is not copyrightable because the menu command hierarchy "was dictated by the nature of the user macros with which it was designed to interact." Supp.Mem.Re: Additional Authority (attached to Docket No. 189) at 9 (quoting *Computer Assocs. Int'l v. Altai, Inc.*, 1992 WL 139364, at \*24, No. 91-7893, slip op. at 50, 1992 U.S.App. Lexis 14305, at \*67 (2d Cir. June 22, 1992)). An implicit premise behind this argument is that the menu command hierarchy was designed to fit the macros. The subtle suggestion is that the macros came first—that they were pre-existing. Necessarily implicit in Borland's argument is the assertion that neither the final version of the menu command hierarchy nor any substantial part of it was preexisting when the macro language was created.

The thrust of the discussion of "compatibility" in *Computer Associates*, however, relies upon *proof* that what the program was designed to fit was already in existence before the program was designed to fit it. Thus, a program designed to fit hardware specifications cannot be protected by copyright unless the program contains expressive elements not substantially dictated by the hardware. Similarly, a program designed to interact with preexisting software, such as the operating system at issue in *Computer Associates* is not entitled to protection to the extent that it is constrained by the need for compatibility with the preexisting software. Thus, the rule makes sense if the premise of a preexisting functional limitation is valid.

In this case, however, there is a very simple answer to the question "Which came first?" The Lotus 1-2-3 interface—or at least a version of it—was written first. All user macros derive from it. Thus, Borland is simply wrong factually to argue that the 1-2-3 interface was constrained by the macros. On this issue, there is no genuine dispute of fact.

I assume in Borland's favor that, like the chicken and the egg, the macro language (as opposed to macros defined by

users using the macro language, which necessarily came later) evolved simultaneously with the menu commands that delimit it. Nevertheless, it is beyond dispute that the macro language did not evolve *first*.

Borland has argued, also, that the need to ensure that the menu commands in any given menu begin with a different letter is a functional constraint. Even that argument fails, however, because of the availability of other symbolic tokens; e.g., A, B, C, . . . or 1, 2, 3, . . .

It is no doubt true that the macros have functional significance. Moreover, as this court found in *Paperback*, the menu "system" is a fundamental part of the functionality of the macro language and the macros. As the Second Circuit has recognized, however, the fact that a form of expression takes on functional character does not remove it from the protection of copyright. *Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1147 (2d Cir.1987); see also *Paperback*, 740 F.Supp. at 58.

Lotus used the slash character ("/") to initiate a command sequence; the character "R," to specify the command "Range"; the character "F," to specify the command "Format"; and the character "C," to specify the command "Currency." A user inputting the characters "/", "R," "F," and "C" in sequence (and supplying necessary additional parameters) will cause the computer to perform the operations associated with the specified commands.

If the keystroke sequence "/RFC—" were stored in a spreadsheet cell that had been assigned the macro "\C," then a user inputting the "\C" keystroke sequence would cause the computer to perform the same operations as would be invoked by the six-keystroke sequence "/", "R," "F," "C," "Enter," "Enter." Because the "\C" keystroke sequence is invoked by pressing the "c" key with the "Alt" key depressed as one would use a "Shift" key, it involves two keys but perhaps only one "keystroke." Thus, depending upon one's definitions, a saving of either five or six keystrokes would result.

Had Lotus preferred, it could have chosen, for example, the command "Scope" instead of "Range," "Appearance" instead of "Format," and "Money" instead of "Currency." Then the

user would invoke the sequence "/", "S," "A," "M" to establish a "money" appearance (\$xxx.xx) for a scope of cells. In that case, the parallel to Borland's argument would be that the existence of "/SAM" macros in users' files (spreadsheet files with cells defining macros may be saved for future use) "dictated" the use of the "Scope," "Appearance," and "Money" commands in the menu command hierarchy. Obviously, the argument is without merit—"Range," "Format," and "Currency" are demonstrably acceptable choices. The fact that Lotus chose one command set for its first version may have made it "necessary" that future versions adopt the same command set; however, the initial choice of the command set was a free choice.

It bears repeating here that Lotus (though not Borland) was entitled to incorporate in version 2.01 the menu command hierarchy that it employed in earlier versions of 1-2-3. Thus, Borland's contention that the macros preceded and dictated the menu command hierarchy of version 2.01 fails because that fact is irrelevant. This is so because Lotus' exercise of creative expression that is at issue in this case (the menu command structures of versions 1A and 2.0) manifestly preceded the users' macros that incorporate the 1-2-3 command hierarchy.

It may be argued that the macros, by themselves, are functionally dictated by the 1-2-3 menu command hierarchy and so are not copyrightable independently of the copyrightability of the menu command hierarchy. Moreover, even if some macros are copyrightable, it may be that the owner of any copyright in a macro is the user who authored the macro and not Lotus. I am not faced with those questions in this case. I conclude only that Borland's argument rests on a false proposition. Thus, to decide this case I need not and do not decide whether Borland is prohibited from reading and interpreting the macros that have been created by users of 1-2-3. Had Borland created a program that read users' 1-2-3 macros and converted them to macros for use in the Quattro programs' native modes, so that they could be interpreted, executed, modified, debugged, etc. by resort to Borland's

command hierarchy, that would have presented a different case from the one now before me.

Borland did not obtain the right to expressive aspects of Lotus' command hierarchy merely because—if it be the case—the 1-2-3 program revolutionized the spreadsheet market. The menu command hierarchy has a functional aspect when incorporated into the keystroke sequences and macros. That functional aspect is separable from the expressive aspect that preceded it. Borland cannot obtain the right to use the macros and keystroke sequences just because the only means of doing so is by infringing expressive features of the Lotus 1-2-3 macro language and keystroke sequences.

The error of Borland's argument may be demonstrated by a simple hypothetical. If an uncopyrighted movie is made from a copyrighted novel (under the authority of an appropriate license), the public is free to copy all aspects of expression *unique to the movie*; however, the novel does not enter the public domain. Similarly, if the macros have uncopyrightable functional aspects, Borland does not infringe the copyright in Lotus 1-2-3 version 1A by duplicating the functionality in any way that does not copy the expressive elements of Lotus 1-2-3 version 1A, but it must not infringe upon expressive elements.

Borland's argument that *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879), establishes a different rule that controls this case is rejected. This is not a case like *Baker v. Selden* in which the system depends on the use of the copyrighted matter. Borland has, in fact, designed a system (Quattro Pro's native mode), using macros and keystroke sequences and using an alternate command hierarchy, that is fully functional.

#### D. More on *Computer Associates v. Altai*

Some further points regarding the Second Circuit's opinion bear upon its applicability to issues in the present case.

### 1. *Beyond Program Code to Nonliteral Expression.*

The reasoning underlying the Second Circuit's "Abstraction-Filtration-Comparison" test extends beyond program code to nonliteral expression. That is, although the issue in that case concerned program code and its structural aspects, the Second Circuit based its conclusions on reasons of broader application. In fact, the court explicitly approved determinations with respect to the copyrightability of certain nonliteral, noncode (nonstructural) aspects of the 1-2-3 spreadsheet in *Paperback*. See *Computer Assocs.*, 1992 WL 139364, at \*16, No. 91-7893, slip op. at 34, 1992 U.S.App. LEXIS 14305, at \*46. Thus, from statements made in the Second Circuit's opinion, albeit *obiter dicta*, I draw support for my conclusion that certain expressive elements of the 1-2-3 user interface may be protected by copyright.

### 2. *Criticism of Whelan.*

Borland's argument that the Second Circuit, with one stroke, knocked out the "conceptual underpinnings" of *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir.1986), *cert. denied*, 479 U.S. 1031, 107 S.Ct. 877, 93 L.Ed.2d 831 (1987) and of *Paperback* cannot be sustained. First, *Whelan* remains good authority in the Third Circuit and may provide guidance for this court in the same way as the Second Circuit's opinion in *Computer Associates*. That is, for courts in the First Circuit, each of these decisions from other federal circuits is instructive and neither is controlling beyond the persuasive force of its reasoning. Second, and more to the point, the Second Circuit criticized the *Whelan* decision for reasons that in large part do not apply to the rulings I made in *Paperback* or to the rulings I have made and now make in this case.

The Second Circuit was critical of *Whelan* on two principal grounds—that it failed to recognize that programs may have more than one idea, and that it mistakenly asserted that all that is not idea is expression. To resolve those problems, the Second Circuit formulated its "Abstraction-Filtration-Comparison" test.

In both these respects, the rulings I have made in the present case are compatible substantively (though not entirely in methodology) with the Second Circuit's views rather than those of the Third Circuit, where those two circuit's differ. For example, as indicated in the *Paperback* opinion, I recognize that the 1-2-3 user interface can be described as incorporating more than one idea. Thus, the *Paperback* opinion identifies the idea of an electronic spreadsheet, the idea of having a readily available method of invoking the menu at command system (which I concluded merged with its expression as the "/" command), and the idea of the two-line moving cursor menu, among others.

### 3. *Baker Not Alone Controlling.*

In its most recent submissions, Borland once again insists that 17 U.S.C. § 102(b) and *Baker v. Selden* control this case, now asserting that the Second Circuit decision supports Borland's argument. Once again, Borland overstates the point. Borland is correct in observing that the Second Circuit treats *Baker* as the "starting point" in analyses of utilitarian works, including computer programs. That view is compatible with my earlier Memorandum and Order. See *Borland*, 788 F.Supp. at 92-93. Borland fails to note, however, that the Second Circuit went on to state:

While *Baker v. Selden* provides a sound analytical foundation, it offers scant guidance on how to separate idea or process from expression, and moreover, on how to further distinguish protectable expression from that expression which "must necessarily be used as incident to" the work's underlying concept.

*Computer Assocs.*, 1992 WL 139364, at \*11, No. 91-7893, slip op. at 24, 1992 U.S.App. LEXIS 14305, at \*32. The Second Circuit's decision cannot fairly be characterized as holding that *Baker* controlled the outcome in *Computer Associates*. The Second Circuit was sensitive not only to its duty of fidelity to precedent, but as well to its duty of fidelity to congressional mandates that came into existence long after *Baker* was decided.

#### 4. "Substantial Similarity" and "Copyrightability."

The Second Circuit referred to its test as one of "substantial similarity" and did not use the term "copyrightability" for any part of the test. I do not understand this difference of terminology to have substantive implications, however, and more especially, not for this case. Nor do I understand the Second Circuit's three-step test to be meant as a rigid barrier to alternate methods of analysis and decision. The court "advised" rather than mandated that the courts within the Second Circuit adhere to the three-step test announced. Moreover, a court in the First Circuit must take account of some degree of dissonance between the First and Second Circuits regarding the methodology of infringement analyses. *Compare Computer Assocs., supra, with Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600 (1st Cir. 1988). Although *Concrete Machinery* outlines a methodology different from the three steps of the Second Circuit's test, I do not understand the First Circuit to have mandated the order of analysis it described. *See Borland*, 788 F.Supp. at 86. I adhere to the view stated in my Memorandum and Order of March 20 that, under existing precedents, prudential concerns about case management may have a bearing on the order in which a court proceeds with its analysis in a complex copyright infringement case. *Id.* Thus, I conclude that I may and should proceed to a copyrightability analysis at this time even though some potentially material factual questions are still unresolved—facts regarding the long prompts, and facts regarding the extent to which the menu commands and command hierarchy may have been dictated by functional considerations.

#### E. Applying the Standard

##### 1. The First Step.

FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to

the most particularized, and choose some formulation, some conception of the "idea," "system," "process," "procedure," or "method"—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

One may describe a number of conceptions of the 1-2-3 user interface. A non-exclusive list, commencing with the most abstract and moving toward the particular, includes:

(1) Lotus 1-2-3 is an electronic spreadsheet.

(2) It is a menu-driven electronic spreadsheet.

(3) Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command.

(4) Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy.

(5) Finally, one may conceive of the interface as that precise set of menu commands selected by Lotus, arranged hierarchically precisely as they appear in 1-2-3. Under this conception, the interface comprises the menu of commands "Worksheet," "Range," "Copy," "Move," "File," "Print," "Graph," "Data," "System," and "Quit," linked by operation of the command "Worksheet" to the menu of commands "Global," "Insert," "Delete," "Column," "Erase," "Titles," "Windows," "Status," and "Page," etc. (The completion of this proposed statement of the "idea," listing all of the more than 400 commands for which "etc." stands, would require several dozen more lines of text.)

Borland argues that the appropriate conception of the "idea" of the 1-2-3 interface is the fifth option. If that were the case, of course, there would be no elements of expression in the menu commands and menu command hierarchy and therefore no copyrightable aspects in them. The premise of Borland's argument is that an "idea" of Lotus 1-2-3 version 2.01 is complete compatibility with earlier versions of 1-2-3, and more precisely with macros generated for use with earlier versions. Borland argues that the precise menu commands and menu structure are necessary to such functional compatibility. Thus, the argument goes, the entire interface of version 2.01 is a functional system or "idea" and is not copyrightable. This argument is essentially tautological. As applied to any case involving a useful article, an argument of this kind would always define the idea to incorporate all the specifics of the particular expression of that idea in the allegedly copyrightable work. Nothing would be copyrightable under this methodology of analysis. The argument is an attempt to win by definition without focusing at any time on any substantive issue concerning the separation of idea and expression.

To select, at the opposite extreme, the very abstract statement of the idea of 1-2-3 as "an electronic spreadsheet" would be to draw an inappropriately abstract boundary between idea and expression. Thus, I concur in a fundamental principle of the *Computer Associates* opinion and reject the contrary proposition in *Whelan*.

Arguably, my Opinion in the *Paperback* decision, where no sharper focus was essential to the outcome, is consistent with accepting a conception of the idea that falls between the second and third formulations above. See *Paperback*, 740 F.Supp. at 67 (electronic spreadsheet having "menu structure"). In any event, I now explicitly recognize that for decision of the issues now before me the selection of functional operations that the spreadsheet performs must be considered part of the idea of the program. Copyrightability depends on expression distinct from the selection of the set of spreadsheet operations that can be performed.

I conclude that an appropriate conception of the "idea" or "system" of the 1-2-3 interface is the fourth of the five alternative conceptions stated above.

## 2. *The Second Step.*

SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

Does the Lotus 1-2-3 user interface include identifiable elements of expression? For reasons stated below, I conclude that it does.

A very satisfactory spreadsheet menu tree can be constructed using different commands and a different command structure from those of Lotus 1-2-3. In fact, Borland has constructed just such an alternate tree for use in Quattro Pro's native mode. Even if one holds the arrangement of menu commands constant, it is possible to generate literally millions of satisfactory menu trees by varying the menu commands employed.

This may be easily demonstrated. Recall the ten commands that appear in Lotus' main menu: "Worksheet," "Range," "Copy," "Move," "File," "Print," "Graph," "Data," "System," and "Quit." One can imagine an entirely plausible spreadsheet in which the "Worksheet" command has been named, quite naturally, "Spreadsheet." Of course, this might require changing the "System" command to avoid two commands abbreviated "S," perhaps to "DOS." The "Quit" command could be named "Exit" without any other modifications. The "Copy" command could be called "Clone," "Ditto," "Duplicate," "Imitate," "Mimic," "Replicate," and "Reproduce," among others (in some cases requiring modification of other com-

mands in the menu). Additional possibilities include "Output" for "Print," "Draw" or "Chart" for "Graph," "Figures" or "Information" for "Data," "Scope" for "Range," and "Transfer" or "Relocate" for "Move."

Just these potential modifications of the main menu yield over 250 combinations of commands in the main menu with ten distinct first letters. Changes in submenus increase the number of possible menu hierarchies *geometrically*. Since there are dozens of independent submenus, the number of possible menu hierarchies is extremely large.

Borland argues that "[t]o hold that an idea, plan, method or art described in a copyright is open to the public but that it can be used only by the employment of different words and phrases which mean the same thing, borders on the preposterous." Borland's Resp. to Pl.'s Renewed Mot. for Summ. J. (Docket No. 183) at 19 (quoting *Crume v. Pacific Mut. Life Ins. Co.*, 140 F.2d 182, 184-85 (7th Cir.), *cert. denied*, 322 U.S. 755, 64 S.Ct. 1265, 88 L.Ed. 1584 (1944)). This case, however, unlike *Crume*, is not a case in which the system "can be effected solely by the employment of words descriptive thereof." *Crume*, 140 F.2d at 184. Use of just the initial letters of command words (together with long prompts) or of other symbolic tokens would have been a sufficient alternate method of implementing the system. In this case, the command words chosen are not necessary to expression of the system nor are they necessarily incident thereto. See *Computer Assocs.*, 1992 WL 139364, at \*13, No. 91-7893, slip op. at 30, 1992 U.S.App. LEXIS 14305, at \*40.

Lotus argues that a large number of substantially different arrangements (hierarchies) could also have been effected. Looking again at just the main menu, is there any reason that the commands "Copy" and "Move," for example, could not have been arranged in the opposite order? Borland argues that the arrangement was necessary, citing evidence that Lotus arranged the menu commands in order of the expected frequency of use.

It is clear that certain command words have been grouped according to function; e.g., there are eighteen commands that

affect the display mode of spreadsheet cells that are grouped together under the "Format" command. Thus, a jury could find that at least some aspects of the arrangement of command words, as opposed to the specific choice of command words, was guided by functional concerns.

This is, however, a disputable fact question that may affect only the scope of relief in this case. See *ABKCO Music, Inc. v. Harrisongs Music Ltd.*, 508 F.Supp. 798 (S.D.N.Y.1981) (court awarded damages based upon contribution to success of infringing work of copyrighted material) (for subsequent history, see 944 F.2d 971 (2d Cir.1991)). I conclude that it cannot be genuinely disputed that a large part of the structure and arrangement of the menu commands is not driven entirely by functional considerations. There are sufficient non-functional aspects that at least hundreds and perhaps thousands of different expressions of the function were possible when Lotus chose the particular structure of menu commands incorporated into Lotus 1-2-3.

This may be demonstrated by examining more closely Borland's argument that the menus were arranged according to the predicted frequency of use of the commands. I assume the truth of Borland's assertion that Lotus predicted before marketing its spreadsheet that the "Copy" command would be used more often than the "Move" command. Nevertheless, that is merely a prediction of frequency of use. It did not require Lotus to list "Copy" before "Move." A user can type a "C" or an "M" with equal ease no matter which command is listed first. If a user prefers to invoke a command by first highlighting it and then typing "Enter," "Move" is only one keystroke from "Copy"; moreover, the same "right arrow" key that takes the cursor from "Worksheet" to "Copy" moves the cursor from "Copy" to "Move." Thus, the order in which commands are listed in a menu has very limited functional value.

In addition, a prediction of frequency of use depends upon who is the predicted user and what the predicted uses are. For example, a user may work on a spreadsheet without printing any of the work performed on that day. Yet, the user will ordi-

narily invoke the "Quit" command appearing at the end of the main menu at the end of each session. Nevertheless, the "Print" command appears before the "Quit" command on the menu.

Many users may rarely invoke the "System" command available in 1-2-3. That is, the need to invoke a disk operating system ("DOS") "shell" from within the spreadsheet program may be for most users of 1-2-3 a rare event. If the commands in the main menu of 1-2-3 are listed in order of predicted frequency of use, why does the "System" command "precede" the "Quit" command? In fact, what does it mean to say "precede"? "Q" neither precedes nor follows "S" on the keyboard. "System" precedes "Quit" as one moves from left to right within the main menu using the "right arrow" key. It is also true that "Quit" precedes "System" as one moves from right to left using the "left arrow" key. The 1-2-3 menus are circular (in jargon, they have a "wraparound" feature)—moving the cursor one step beyond the "end" of the menu causes the cursor to come to rest at the opposite "end." Thus, from the default cursor position on the "Worksheet" command, arguably "Quit" precedes "System."

For all these reasons, any *ex ante* prediction of frequency of use is itself of limited usefulness. It follows that the arrangement of menu commands according to predicted frequency of use is not a major functional limitation on the number of arrangements of menu commands.

The menu command hierarchy is an integral part of the functionality of the macros and of the keystroke sequences. Nevertheless, the fact that the macros and keystroke sequences incorporate the menu command hierarchy into their functionality does not remove the menu command hierarchy from the scope of copyright, if otherwise subject to copyright protection. Moreover, the macros and keystroke sequences are protected to the extent that it is necessary to infringe a copyright to use them. Of course, as I have stated above, it was *not* necessary to copy expressive aspects of the macro language and keystroke sequences to copy their function.

### 3. *The Third Step.*

THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work."

The question posed by this element of the copyrightability test is whether the creativity involved in establishing the menu commands, menu command hierarchy, macro language, and keystroke sequences was more than trivial. No reasonable jury could find otherwise. As Borland has itself acknowledged, at least implicitly, Lotus 1-2-3 was a dramatic change and improvement over what was available on the market at the time Lotus was created. Although a large portion of that improvement relates to the functional aspects of Lotus 1-2-3, the features that I have now concluded are expressive also played a substantial role. Borland has maintained that those features are part of an uncopyrightable system (an argument I now reject for the reasons stated), but Borland has never argued that they were trivial, nor could it do so persuasively.

#### *F. Ashton-Tate v. Ross*

Borland argues that the Ninth Circuit's decision in *Ashton-Tate Corp. v. Ross*, 916 F.2d 516 (9th Cir.1990), *aff'g* 728 F.Supp. 597 (N.D.Cal.1989) militates in favor of a conclusion that the menu commands and command hierarchy are not copyrightable. The Ninth Circuit held that a list of menu commands was not copyrightable for reasons stated by the district court. The district court, in turn, held that a document bearing "a list of labels for user commands, many of which are common commands that were already available on other software programs" was not innovative or novel. *Ashton-Tate Corp. v. Ross*, 728 F.Supp. 597, 602 (N.D.Cal.1989). The relevance of that conclusion to this case is in some doubt in view of the fact that Lotus 1-2-3 is one of the programs on which

the commands were already available. There is no evidence in this case that the commands available in 1-2-3 were common commands at the time of Lotus' authorship.

The district court also held that there was nothing innovative in the order in which the commands were listed. *Id.* Borland has submitted in this case a copy of the list of commands at issue in *Ashton-Tate*. Borland's Mem. in Supp. of Cross-Mot. for Summ. J. (Docket No. 141) at 85-86. That list contains well under one hundred commands scrawled on one sheet of paper segregated into under one dozen functional groups. In form, detail, arrangement, and content, it bears almost no resemblance to the hundreds of menu commands arranged in Lotus' dozens of menus. The relevance of the court's conclusion to the case at bar extends, if at all, only to the copyrightability of subelements of the 1-2-3 menu hierarchy, and depends upon factual inquiries that remain to be made.

Finally, the court concluded that in the absence of any contribution to the user interface other than the command list (including, especially, any contribution to the program code), the contribution of the command list was a mere contribution of ideas. As Judge Learned Hand observed, however, and many others have agreed, decisions in which a line is drawn between idea and expression have an ad hoc character. They tend to be fact sensitive and case specific. The absence of other copyrightable contributions in *Ashton-Tate* may have caused the court to reach a determination in that case with respect to the copyrightability of menu commands that it might not have reached had the defendant contributed other copyrightable elements.

In the interest of completeness and candor, I note as well that courts in one circuit are not bound by the decisions of other circuits. I view my obligation as one of determining the law manifested in the Copyright Act as it would be interpreted and applied to this case by the Court of Appeals for the First Circuit and the Supreme Court should this case reach either or both of those courts.

#### IV. SUBSTANTIAL SIMILARITY (ILLICIT COPYING)

For the reasons stated in Part II, above, and supplemented briefly here, I cannot conclude that Borland has copied substantially the whole of the Lotus 1-2-3 user interface.

In *Paperback*, although witnesses called attention to "look and feel" as a descriptive metaphor, counsel for Lotus did not base their contentions on this metaphor, and the court in its decision did not rely upon it. 740 F.Supp. at 62-63. One reason the metaphor did not seem useful in that case is that the concept of "look and feel" relates more to substantial similarity (in the mixed law-fact sense) than to copyrightability. In *Paperback*, so much of the 1-2-3 user interface had been copied that it was not difficult to resolve questions of substantial similarity in the mixed law-fact sense. The difficult questions in that case centered only around copyrightability of the user interface.

*Paperback* involved an appropriation by the defendant of the entire "look and feel" of 1-2-3. In this case, Borland has appropriated, to a great extent, the "feel" of the 1-2-3 user interface and only to a lesser extent the "look" of 1-2-3. Indeed, Borland has designed an interface that in many respects looks substantially different from the 1-2-3 user interface. Borland's colors and pull-down menus are but two examples of the differences in "look" in the Quattro programs. Of course, the menu commands and the menu command hierarchy look the same in both programs.

The "feel," on the other hand, of the emulation modes of the Quattro programs depends in large part on the keystroke sequences one enters to perform spreadsheet operations. One enters the same keystroke sequence to perform the same spreadsheet operations in both 1-2-3 and Quattro Pro's emulation mode. They feel the same. Thus, an experienced user accustomed to the 1-2-3 interface needs to look seldom, if at all, to achieve the desired result in the emulation modes of the Quattro programs.

This conclusion may be expressed in a more straightforward way without use of any part of the "look and feel metaphor." Lotus is entitled to a judgment of infringement only if Borland appropriated copyrightable elements of 1-2-3 and those copyrightable elements, taken together, make relevant portions of Borland's program substantially similar to 1-2-3. The evaluation of substantial similarity (in the mixed law-fact sense) therefore depends upon determining what copyrightable elements of the Lotus 1-2-3 user interface Borland copied. Even if I assume, however, that Borland did not copy the long prompts, and that some aspects of the menu commands, menu command hierarchy, macro language, and keystroke sequences of 1-2-3 are not copyrightable, I conclude that no reasonable jury, applying the law, could find other than that the Quattro programs infringe 1-2-3. That is, a reasonable factfinder must conclude that the Quattro programs derive from illicit copying. The emulation interfaces are substantially similar in the mixed law-fact sense to the Lotus 1-2-3 user interface. (Returning to the metaphor, one may say that is why they "feel" the same.)

I am not able to determine on motion for summary judgment the precise scope of Borland's infringement. Genuinely disputable factual questions exist that may affect the scope of substantial similarity and therefore the nature and scope of the remedies for infringement in this case. Nevertheless, it is clear that Borland has infringed the 1-2-3 user interface, at least in substantial part.

Lotus argues that questions of substantial similarity in the mixed law-fact sense ought not to be put to a jury. I understand Borland to contend otherwise (though it does not explicitly address the point in its latest submissions, perhaps because the Memorandum and Order of March 20 may have left the impression that I would submit such questions to a jury if a genuine dispute of fact existed). The scope of issues to be put to the jury is significantly reduced by my ruling that the emulation modes of the Quattro programs have a core that is substantially similar to 1-2-3. Borland's request that I put questions of fact concerning substantial similarity in the

mixed law-fact sense to a jury other than the one I empanel to hear evidence regarding substantial similarity in the evidentiary sense is now moot.

Borland's Quattro programs infringe Lotus 1-2-3 because the extent of copying of copyrightable elements of 1-2-3 renders the Quattro programs substantially similar to 1-2-3. If the jury finds that Borland also copied other copyrightable elements, then the Quattro programs will, as a matter of law, be even more substantially similar to 1-2-3; that is, the scope of Borland's infringement will necessarily be broader. For example, if the jury finds that Borland copied the Lotus 1-2-3 long prompts, and if I conclude that the long prompts are copyrightable expressive elements of 1-2-3, those two determinations, without any further finding that Borland's long prompts are substantially similar to Lotus' long prompts in the mixed law-fact sense, support a conclusion that Borland's Quattro programs infringe the long prompts of 1-2-3. There would be no need to ask separately whether the copying of the long prompts would alone, or in combination, render the Quattro programs substantially similar to Lotus 1-2-3. Substantial similarity (in the mixed law-fact sense) is determined by comparing the copied copyrightable elements of the infringing work *all together* with the copyrighted work as a whole. I have already determined that this comparison, even if the long prompts were not copied, requires a determination of infringement.

In reaching these conclusions, I once again reject Borland's contention that it is entitled to place in evidence at trial all the elements of *Quattro Pro* that were not copied from Lotus. Despite Borland's otherwise detailed (if not precisely accurate) explication of the *Computer Associates* case, Borland has failed to acknowledge an express conclusion of the Second Circuit that is contrary to Borland's position:

[I]n some cases, the defendant's program structure might contain protectable expression and/or other elements that are not found in the plaintiff's program. Since it is extraneous to the allegedly copied work, this material would have no bearing on any potential substantial sim-

ilarity between the two programs. . . . Furthermore, by focusing the analysis on the infringing rather than on the infringed material, a court may mistakenly place too little emphasis on a quantitatively small misappropriation which is, in reality, a qualitatively vital aspect of the plaintiff's protected expression.

*Computer Assocs.*, 1992 WL 139364, at \*22, No. 92-7893, slip op. at 47-48, 1992 U.S.App. LEXIS 14305, at \*64-\*65.

That Borland, in developing the Quattro programs, has added functional and expressive elements that do not exist in 1-2-3 is irrelevant in view of the fact that Borland copied virtually the whole menu command structure of 1-2-3 into its emulation interfaces. Borland's additions have caused some variation in the manner in which the elements taken from 1-2-3 are expressed in the Quattro programs. For example, in the main menu, the "View" command, a command not present in the main menu of 1-2-3, is interposed between the "System" and "Quit" commands. A decisionmaker in this case (whether judge or jury) must ignore the added expression *to the extent that it does not change the expression Borland copied from Lotus*. I conclude that no reasonable jury could find for Borland that Borland did not take the menu commands, menu command structure, macro language, and keystroke sequences substantially as they were.

## V. BORLAND'S DEFENSES

Lotus has renewed its motion for summary judgment that Borland's affirmative defenses fail. Borland has raised three affirmative defenses in this action: waiver, laches, and estoppel. The issue of Borland's affirmative defenses was raised and fully briefed by the parties in conjunction with their previous motions for summary judgment. Because I denied both motions on other grounds by Memorandum and Order of March 20, there was no need to consider Borland's affirmative defenses at that time and I did not do so. Having concluded that Borland has infringed Lotus' copyright in 1-2-3,

I proceed to address Lotus' motion for summary judgment on Borland's alleged defenses.

### A. Waiver

The parties agree that to succeed in its defense of waiver, Borland must demonstrate that Lotus voluntarily and intentionally relinquished its right to pursue Borland in this action. Borland's argument, however, is based upon *Borland's* "state of mind" and Lotus' alleged *concealment* of Lotus' "state of mind." Borland has not presented any evidence that Lotus manifested to either Borland or the world that it would not pursue Borland in this action. Borland's allegation that Lotus sought to play down the import of the *Paperback* litigation in its public relations campaigns by stating that *Paperback* involved "99% clones" does not amount to waiver, even if proved. Therefore, I grant Lotus' motion for summary judgment on the defense of waiver.

### B. Laches

The parties agree that to succeed in its laches defense, Borland must demonstrate that Lotus delayed unreasonably in bringing this action and that as a result Borland suffered undue prejudice. Borland argues that it expended \$18 million developing the Quattro programs, at least in part because, by not suing earlier, Lotus led Borland to believe that Lotus would not sue. Borland has admitted that it would have developed a spreadsheet program in any event at substantial cost. Nevertheless, Borland argues that the costs of releasing the entire spreadsheet constitute prejudice. This argument is plainly without merit.

Borland also argues that its decision to offer a 123-compatible interface is hard to undo and that it has expended large sums in advertising keystroke and macro compatibility of the Quattro programs with Lotus 1-2-3.

Without deciding at this time issues regarding alleged undue delay, I note that the question of prejudice may depend

upon determinations yet to be made regarding the appropriate remedy or remedies for the kind and scope of infringement that has occurred in this case. I take note also that precedents on laches derive from equity, and I may rule at trial that evidence admissible only in relation to equitable defenses must be received out of the presence of the jury, or in a later phase of trial. I deny Lotus' motion for summary judgment on the laches defense without prejudice to Lotus' renewing its arguments in support of the motion at a later date.

### C. Estoppel

Estoppel, like laches, has its origins in equity. Without finally ruling on the matter at this time, I take note that it may be appropriate either to receive evidence bearing solely on estoppel (or only on estoppel and laches) out of the presence of the jury, or at a phase of trial after that in which issues relating to the scope of infringement are finally resolved.

Borland argues that Lotus' intentional concealment of its belief that the Quattro programs infringe 1-2-3 is a ground for estoppel. Borland's argument rests on a premise borrowed from an explicit statutory requirement in the law of patents, that Lotus owed a duty to Borland to notify Borland of its infringement. The law of copyright has not imported that requirement. Thus, Borland's arguments that Lotus concealed its intentions and that Borland did not and could not have known Lotus would sue are meritless. Lotus 1-2-3 is copyrighted. Borland copied copyrightable elements of 1-2-3 that constitute a substantial part of that program. Lotus has sued, and Borland is liable.

I reject as well Borland's argument that Lotus' delay in filing suit is a ground for estoppel. That argument is merely a repetition of the argument Borland raised in support of its laches defense.

Borland's identification of the alleged Lotus misrepresentation on which Borland relied to its detriment is less than pellucid. Borland refers to affirmative indications by Lotus that it would not sue, but fails to identify what indications Borland alleges, if any, other than comments made for public

relations purposes and reported in trade publications and position papers. Borland says, for example, that Lotus characterized the *Paperback* litigation as involving "99% clones." Borland argues that the Quattro programs are not clones of 1-2-3 and that it reasonably came to believe, therefore, that Lotus would not sue. If this is the best argument Borland can advance for justifiable reliance, the likelihood that it can present an issue on estoppel for a factfinder (judge, if this is treated as an issue in equity, and jury otherwise) seems very remote. Though I will not grant Lotus' motion for summary judgment at this time, the Order Regulating Trial in this case will require evidence bearing only on estoppel (or on estoppel and laches) to be proffered first outside the presence of the jury.

### VI. CONCLUSION

It is my conclusion that, as a matter of law, Borland's Quattro products infringe the Lotus 1-2-3 copyright because of (1) the extent of copying of the "menu commands" and "menu structure" that is not *genuinely* disputed in this case, (2) the extent to which the copied elements of the "menu commands" and "menu structure" contain expressive aspects separable from the functions of the "menu commands" and "menu structure," and (3) the scope of those copied expressive aspects as an integral part of Lotus 1-2-3. Nevertheless, I conclude that a jury trial is essential before final disposition of this case because the scope of relief available will depend in part on whether the jury finds for Lotus on disputed factual contentions that the copying of separable expressive elements of the Lotus 1-2-3 user interface into the Quattro programs was greater than the minimum essential to constituting a substantial part of the Lotus 1-2-3 work.

In summary, Lotus' Motion for Summary Judgment is allowed in part and denied in part, as stated in the foregoing rulings. Borland's motion for summary judgment is denied.

Having reached this conclusion, I will state these rulings in the form of Determinations of Undisputed Facts and Con-

clusions of Law incorporated as a section of a draft Order Regulating Trial, to be distributed to counsel separately from this Memorandum and Order. An Order Regulating Trial will be entered at or after the Pre-Trial conference scheduled for September 23, 1992. Each party is invited to submit, by Memorandum filed on or before September 1, 1992, proposed modifications, deletions, and additions to the draft Order Regulating Trial. Responses may be filed on or before September 15, 1992.

UNITED STATES DISTRICT COURT  
D. MASSACHUSETTS

Civ. A. No. 90-11662-K

March 20, 1992

---

LOTUS DEVELOPMENT CORPORATION,

*Plaintiff,*

—v.—

BORLAND INTERNATIONAL, INC.,

*Defendant.*

---

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, Mass., Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, Mass., Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, Cal., David L. Hayes, Fenwick & West, Gary L. Reback, Wilson, Sonsini, Goodrich & Rosati, Mitchell Zimmerman, Fenwick & West, Palo Alto, Cal., Peter Erich Gelhaar, Donnelly, Conroy & Gelhaar, Boston, Mass., *for defendant.*

---

MEMORANDUM AND ORDER

KEETON, *District Judge.*

In this civil action, the plaintiff, Lotus Development Corporation ("Lotus"), seeks damages and equitable relief for alleged infringement by defendant, Borland International, Inc. ("Borland"), of the Lotus copyright in its computer software

program, Lotus 1-2-3. This is the same copyright for the infringement of which Lotus has obtained relief under this court's decision in *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass. 1990) (holding that Paperback's VP Planner was an infringing software product).

Lotus contends that on undisputed facts Borland's Quattro and Quattro Pro (herein, collectively, the "Quattro programs") are likewise infringing. Borland responds that its products are materially different from both Lotus 1-2-3 and VP Planner and that the court should determine on undisputed facts that the Quattro programs do not infringe the Lotus copyright. Borland argues that this result follows under the rules and standards of law applied by this court in *Lotus v. Paperback*. In the alternative, Borland contends that this court should reconsider its rulings of law in the *Paperback* case and now hold that the elements of Lotus 1-2-3 it determined previously to be copyrightable are not copyrightable. Also, Borland contends that in any event it has valid defenses that Paperback did not have.

Pending before the court are cross-motions for summary judgment, fully briefed and with factual submissions complete except for confidential materials to be presented to the court pursuant to a stipulation and protective order.

For the reasons explained in this Memorandum, I conclude that neither party's motion is supported by the record now before the court.

The primary contention of Lotus that Borland copied the Lotus user interface as a whole fails because there is a genuine dispute of fact as to whether Borland did so. Borland's primary contention that it is entitled to a summary judgment on copyrightability fails on legal grounds for reasons explained in Part VI of the Memorandum.

Each party has advanced alternative contentions for summary judgment. Each of the submissions before me, however, fails to meet the movant's burden of identifying both a clearly stated legal theory and a clear statement of undisputed facts sufficient to demonstrate an entitlement to summary judgment. In these circumstances, I conclude that it is appropriate

to deny the cross-motions for summary judgment. Explicitly, however, I do not preclude the filing of another motion by either party if one can be presented consistently with the rulings stated and explained in this Memorandum.

### I. Redacted Submissions

Redacted submissions (Docket Nos. 141 and 147 and certain affidavits, declarations, and exhibits) now before the court were filed pursuant to this court's practice (for the protection of parties, counsel, and court personnel from risks of inadvertent disclosure) not to allow protected materials to be filed with the Clerk before the judge (not any other person acting under a delegation of authority) has approved in writing the specific filing.

In this instance, I have concluded that the description of the protected materials, appearing in the redacted filings, is sufficient for me to determine that they are not material to any of the rulings stated and explained in this Memorandum. If counsel think otherwise, however, they may, with notice to the court and opposing counsel, bring the protected materials to the next scheduled conference to be presented to the court for examination during that conference. In light of this ruling, the parties' motions regarding the submission of confidential documents (Docket Nos. 140, 156, and 160) are dismissed as moot.

### II. Disposition of the Cross-Motions for Summary Judgment

This court held in *Paperback* that the 1-2-3 user interface, taken as a whole, was copyrightable. That ruling was never appealed or vacated and continues to have the limited precedential force of a district court decision. The applicability of that precedent to this case is disputed, however, because Borland contends that its allegedly infringing products are materially different from VP Planner (the computer work involved in *Paperback*). VP Planner has been described, both in and

outside this court, as an imitation (or look-alike, or clone) of 1-2-3. The conclusion that the user interface of 1-2-3, as a whole, was copyrightable was outcome determinative as to a computer work like VP Planner because, without dispute, the 1-2-3 user interface as a whole was copied. Lotus has maintained, however, both before this court and elsewhere that there is a distinction between imitations of 1-2-3 and other products such as Microsoft Excel.

In the present case, unless other issues are dispositive, it will be necessary to determine on which side of an unmarked boundary, between imitations of 1-2-3 and products such as Excel, the Quattro programs fall. I cannot determine on the record before me that the "1-2-3 interface" (also called the "emulation interface") of the Quattro programs is indisputably an imitation of the 1-2-3 user interface. Thus, for Lotus to meet its burden of showing entitlement to a summary judgment of infringement it must (a) identify expressive elements in 1-2-3 that were indisputably copied in the Quattro programs, (b) establish that those expressive elements, either separately or together, are as a matter of law copyrightable, and (c) establish that the copied expressive elements of the Quattro programs' emulation interfaces are substantially similar to copyrightable elements of the 1-2-3 interface.

The conclusion that the user interface as a whole is copyrightable (which this court reached in *Paperback*) does not resolve the further questions that may now have to be resolved regarding the copying and copyrightability of individual parts or a sum of parts less than the user interface as a whole. The Lotus motion and supporting submission fail to focus precisely, even in the alternative, any claim of copyrightability and undisputed copying of something less than the user interface as a whole. In these circumstances, having concluded that the claim of copying the user interface as a whole has not been demonstrated, I must conclude also that Lotus has failed to demonstrate it is entitled to summary judgment.

Similarly, Borland has failed to demonstrate support for its contention that, even if it copied something from 1-2-3, nothing Borland copied was copyrightable. As far as I am now

able to determine, a factfinder may find on disputed evidence that Borland copied the 1-2-3 user interface as a whole. I have previously determined, and as explained in this Memorandum now adhere to the view, that the 1-2-3 user interface as a whole is copyrightable. Moreover, the present record suggests that a factfinder surely could find that Borland copied some expressive elements of the 1-2-3 user interface. The outcome of this case may depend on a more precise focus than Borland has presented as to what elements Borland copied and whether one, or more, or some set or sets of those elements are copyrightable.

One effective way of focusing contentions and considering whether they present genuine disputes of fact that are material under the court's resolution of contested issues of law is to consider how any fact questions might be framed for the factfinder's consideration. Especially is this so when a demand for jury trial has been filed. Questions to be submitted to the jury must at some point during full preparation for trial be framed precisely. The trial judge and trial lawyers may well commence this task early, to aid as well the focusing of arguments on motions for summary judgment. A lawyer who contends that a genuine dispute exists as to some material fact should be able to frame clearly a "written question[ ] susceptible of categorical or other brief answer," proposed for use in a "special verdict," Fed.R.Civ.P. 49(a). If, when challenged to do so, counsel cannot state precisely any proposed question of fact that is disputed and material, summary judgment is almost certainly appropriate. If, on the other hand, counsel can frame even a single disputed and material question of fact, summary judgment is inappropriate.

On the present submissions, I am not able to conclude that either party has met its burden of showing a basis for summary judgment in its favor. I therefore deny the motions at this time; however, I conclude that it is appropriate to allow the parties an immediate opportunity to focus their arguments more precisely, on the chance this may lead to disposition without the high costs of trial, and in any event will better focus issues for trial if a trial is necessary. Therefore, I deny

the motions now before me, but will allow each party, on the schedule stated below, to file a new, better focused motion if the party concludes it can surmount the serious obstacles identified in this Memorandum.

The claims and defenses in dispute in this case present issues of law and issues of fact that are independently complex. The relationships among the law and fact issues dramatically enhance complexity. A jury demand and the resulting need for clear and explicit explanations to jurors of the law they need to understand in order to identify and resolve material disputes of fact produce still more consequences of significance, and perhaps more complexity as well. Complexities and their consequences have significant implications for fundamental issues of fair adjudication. They present also some pragmatic issues of case management in both pretrial and trial proceedings. The Order entered at this time is aimed at creating a context in which the parties and their attorneys will have incentives to use their adversary presentations in ways that will better focus and illumine issues for the decisionmakers, judge and jury, and will save the parties, their counsel, and the public from needlessly excessive litigation costs. Suggestions of counsel in aid of this aim are invited and encouraged.

### *III. Entanglement of Law and Fact as to Copyrightability and Substantial Similarity*

#### *A. Basic Methods of Separating Law and Fact*

In general, an issue of law is decided by a court—not by a jury in a jury trial and not by a trial judge as factfinder in a nonjury trial. Also, in general if reasonable persons can differ about how a fact question should be decided on the evidence received in the trial, the jury decides it in a jury trial, and the jury decision is final; in a nonjury trial, the judge as factfinder decides, and the decision is subject to review under a deferential standard. Fed. R.Civ.P. 52(a) (findings sustained unless “clearly erroneous”).

These clear and straightforward rules sometimes require supplementation, however, when the issues of law and fact are “mixed.” If a court can clearly separate law from fact, it may use either of two basic methods of submitting questions to a jury. One method asks the jury to answer a mixed law-fact question; the other, a strictly factual question.

Under the first method, the court submits a combined law-fact question to the jury. To use this method properly the court must first decide the issues of law so it can give the jury clear instructions on the law. The jury’s answer, if they discharge their responsibility well and truly, decides only fact questions even though their answer is in form an answer to the mixed law-fact question.

This first option may be used either for a “special verdict” under Rule 49(a), or for a “general verdict accompanied by answers to interrogatories” under Rule 49(b) of the Federal Rules of Civil Procedure.

The second option open to the trial judge is to frame “special questions” for the jury under Rule 49(a), or “interrogatories” under Rule 49(b), that ask only fact questions, not mixed law-fact questions.

A charge that directs the jury to return only a general verdict cannot use the second method because the general verdict is necessarily a mixed law-fact finding (the type required under the first option). Thus, a verdict “for the plaintiff” or “for the defendant” does not provide any basis for a reasoned determination by the court that the verdict rests on findings of fact supported by evidence and not on a different view from that of the court about the legal elements of mixed law-fact determinations.

If only a “special verdict” under Rule 19(a) is to be returned, and the questions are well framed, the jury need not have any explanation of the legal rule or rules the court will apply to determine what judgment to enter in view of the jury’s findings. The judge may choose to explain anyway, but giving the explanation is a matter of choice rather than necessity.

When two or more questions of fact must be submitted, the court may submit one or more questions by one of these meth-

ods and one or more by the other. Also, the court may use a mixture of general verdict and "interrogatories," Fed. R.Civ.P. 49(b).

When issues of law and fact are easily separable, the court, when it perceives a need to do so to avoid jury confusion or bias, may use a "special verdict" form under Rule 49(a) that strictly submits fact questions; also, the court may cleanly separate the hearings before the court on the legal issues from those before the jury on the fact questions. A key purpose of trying a case in this way is to reduce the risk that the jury will be improperly influenced by evidence and arguments not admissible in relation to the strictly factual issues submitted to them.

When law and fact are not easily separable—regardless of how deep the entanglement may be—it nevertheless remains exclusively the responsibility of the court to decide the legal questions.

Ordinarily, it is the responsibility of the factfinder (jury or, in a nonjury case, trial judge) to decide the fact questions, but exceptions exist, as will be noted in Parts IV, VI, and VII below.

## B. Problems of Entanglement

In order to use a strictly factual special verdict form, a court must cleanly separate law from fact. Not all mixed questions of law and fact can be easily separated. Indeed, there may be instances in which law and fact are so deeply intertwined that, at least as a practical matter if not strictly in principle as well, total separation cannot be achieved. In copyright cases, is "substantial similarity" such an instance? Is "copy-rightability" such an instance?

### 1. Substantial Similarity

"Substantial similarity" is not consistently used with a single, settled meaning. In the law of copyright this phrase is used in at least two distinct senses. For this reason, the use of the term "substantial similarity," absent some contextual guidance, may be very misleading.

As part of its *prima facie* case, a plaintiff must prove substantial similarity between copied copyrightable elements of the copyrighted work and expressive elements of the allegedly infringing work. "Substantial similarity," as used in this statement, indicates a degree of similarity between the allegedly infringing material and what is copyrightable (that is, the copyrightable part or parts). It is not enough for a plaintiff to prove great similarity of the allegedly infringing work to uncopyrightable parts of the copyrighted work.

The similarity to copyrightable parts, it is said, must be such that an "ordinary observer" would find that there has been "unlawful appropriation." *Concrete Machinery Co. v. Classic Lawn Ornaments*, 843 F.2d 600, 608 (1st Cir.1988). Thus, the copying must be extensive enough to be "substantial." This is "substantial similarity" in a mixed law-fact evaluative sense. Thus, the test for determining whether there is "substantial similarity" in this sense involves not merely a ministerial task of measuring by a yardstick or word count, but a judgmental task of weighing the quantitative measurements along with other relevant factors and coming to an overall evaluation that applies a legal test for "unlawful appropriation" to the facts.

The plaintiff must also demonstrate copying of the copyrighted work. Unlike patent law, copyright law never determines that an infringement occurs merely because two works are similar, or even identical, so long as the works are independently created. Therefore, to be successful, a plaintiff must demonstrate that the defendant had access to plaintiff's work. Such access is not disputed in this case. Nevertheless, to prove copying, plaintiff must show more than access. Copying can, of course, be proved directly, and there is some direct evidence of copying in this case. However, once access is proved, copying can also be proved by demonstrating "substantial similarity." In this context, "substantial similarity" simply means sufficient similarity of a given element of a work to an element in the allegedly infringing work to support a reasoned inference that more probably than not the element was copied from the copyrighted work. This is not similarity

in a mixed law-fact sense that includes being similar enough to constitute "unlawful appropriation." Rather, the elements of the copyrighted and allegedly infringing work must be shown to be substantially (i.e., notably) similar in a purely factual sense. This is "substantial similarity" in an evidentiary sense. "Substantial similarity" in this sense is one kind of circumstantial evidence of copying.

In this Memorandum, I will try in each instance to indicate clearly (by explicit statement, by context, or by both) to which usage of substantial similarity I am referring.

## 2. Consequences of Entanglement

The complexity introduced by the entanglement of law and fact, bearing on "substantial similarity" (in the mixed law-fact evaluative sense) and on copyrightability has important consequences, including distinctive problems of adjudication and case management both during pretrial proceedings and during trial. Some of these consequences are relevant to the present case.

First, fashioning appropriate verdict forms and instructions to the jury is more difficult when law and fact are so entangled that the usual mode of stating an issue is in the mixed law-fact form.

Second, determining the scope of evidence that is admissible in relation to questions to be decided by the factfinder (jury or trial judge in a nonjury case) is far more difficult. Moreover, failure of the trial judge to resolve problems of admissibility, especially in relation to proffered expert opinion testimony, may substantially impede fair trial because of risks of jury confusion and the incentives to counsel and experts to try to influence the jury with arguments on the law cast in the form of opinions on the mixed law-fact questions.

Third, when the applicable law requires the application of a standard for decisionmaking that is derived from a congressionally mandated accommodation among conflicting public policy interests and thus "reflects a balance of competing claims upon the public interest," *Sony Corp. v. Universal City Studios, Inc.*, 464 U.S. 417, 431, 104 S.Ct. 774,

783, 78 L.Ed.2d 574 (1984), consequences of the first and second types are enhanced, sometimes beyond measure. Difficult as the task of applying that balance in particular cases may be, however, it is not open to a court to ease the task either by revising that balance, *Feist Publications, Inc. v. Rural Telephone Service Co.*, \_\_\_ U.S. \_\_\_, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991), or by leaving the jury free to revise that balance. A congressionally mandated balance must be respected by courts and juries alike.

The mixed law-fact evaluative "substantial similarity" issue in copyright law inevitably produces the first and second of these types of consequences, and in particular cases may produce the third as well unless effective measures of judicial control are used. For example, consider whether or not expert opinion evidence about the ways in which two computer software programs are similar and dissimilar should be heard by a jury. One may argue for a negative answer under the authority of *Concrete Machinery Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 608 (1st Cir.1988) (substantial similarity is to be judged under the "ordinary observer" test "unaided by expert testimony"). Even if it may be appropriate and useful to a jury in understanding purely historical facts about characteristics of computer programs to hear expert opinions bearing on historical facts, still opinions that explicitly or implicitly depend on policy premises inconsistent with the law (as determined by the trial judge, subject to correction on appeal) should not be heard by the jury.

Affidavits of experts and arguments of counsel before me in this case illustrate the risks of contentious and potentially unfair proceedings incident to the strong tendency of witnesses and lawyers to intertwine every statement about factual similarity with opinions and arguments about what is, or even should be, the kind of similarity that is legally significant. These opinions are, of course, opinions on the law. Allowing a jury to hear them expressed by witnesses and lawyers is fundamentally inconsistent with the rule that jurors take their instructions on the law solely from the court.

I will return to the issue of "substantial similarity" in the mixed law-fact evaluative sense in Part IV, below.

### 3. Copyrightability

Issues of "copyrightability" in computer software programs present the third type of consequence of entanglement of law and fact in a dimension so extreme that the problem is appropriately viewed as one different in kind and not merely degree. This is one reason, though not the only one, for concluding that at least in the present case, and perhaps more generally, all questions of law and fact bearing on copyrightability of elements of a computer software program are to be decided by a court, not a jury. That issue is considered more fully in Parts IV, VI, and VII, below.

### IV. Entanglement of Copyrightability and Substantial Similarity

The complexity of entanglement of issues in this case extends not only to issues bearing only on copyrightability, and to issues bearing only on substantial similarity in the mixed law-fact evaluative sense, but also to the interdependence of copyrightability and substantial similarity.

The point is illustrated by Borland's argument that Borland is entitled to summary judgment *on the issue of copyrightability*, Docket No. 141, at 1 (emphasis added), because the elements of Quattro and Quattro Pro that are alleged to be *substantially similar* to elements of Lotus 1-2-3 concern only the way certain parts of the programs function, not copyrightable expressions. *Id.* at 4-8.

Borland argues that *Concrete Machinery*, 843 F.2d at 600, requires that issues of substantial similarity be addressed first, to determine whether "there are sufficient articulable similarities to justify a finding that the defendant has copied from the protected work." Borland's Memorandum, Docket No. 141, at 114 (quoting *Concrete Machinery*, 843 F.2d at 608) (citing *Arnstein v. Porter*, 154 F.2d 464 (2d Cir.1946)). (In this argument, one may note, Borland is using "substantial similarity" in the evidentiary sense.) Then, Borland argues, "the court must determine whether the copying is sufficiently substantial to constitute 'unlawful appropriation' ('illicit

copying')." *Id.* at 114 (quoting *Concrete Machinery*, 843 F.2d at 608). (This argument, it may be noted, focuses both on a strictly factual question about copying and a mixed law-fact question about whether the copying is sufficient to constitute "unlawful appropriation.") "Assuming copying of protected aspects is established, the trier of fact can then assess pursuant to the 'ordinary observer' test whether there is substantial similarity between the protected expression and the accused work." *Id.* (quoting *Concrete Machinery*, 843 F.2d at 609).

The first step formulated in the *Concrete Machinery* opinion may be perceived as itself composed of two parts—determining what are the "protected aspects of the copyrighted work" (that is, the "copyrightability" issue) and determining whether there has been copying of one or more of those aspects. It is not clear that one of these two need precede the other in an analysis of the evidence and arguments presented to a court. Indeed, they may be sufficiently interrelated that in most cases they are best addressed together—a possibility at least suggested by the First Circuit's combining the two into the first "step" of its "two-step" test.

In any event, the argument that First Circuit precedent requires a particular order of analysis of a given problem is contrary to a substantial body of precedent in the circuit. For example, the First Circuit spoke quite early to make clear, in the context of discrimination cases, that the factors analysis set forth in *McDonnell-Douglas Corp. v. Greer*, 411 U.S. 792, 93 S.Ct. 1817, 36 L.Ed.2d 668 (1973), is intended as a useful guideline and not a mandatory straitjacket. *See, e.g., Dance v. Ripley*, 776 F.2d 370, 373 (1st Cir.1985) ("This circuit, along with other circuits, has rejected the argument that *McDonnell Douglas* and [*Texas Dept. of Community Affairs v. Burdine* 450 U.S. 248, 101 S.Ct. 1089, 67 L.Ed.2d 207 (1980)] set forth a 'rigid, three-step proof process in Title VII cases.'") For like reasons, I conclude that the pattern of analysis and decisionmaking presented in *Concrete Machinery* is not meant to be a straitjacket. Thus, I conclude that the order of addressing the various elements of a *prima facie* case fashioned in *Concrete Machinery* need not be applied to a case in which

there are compelling practical reasons for a different order of proceeding.

Perhaps the procedure discussed in Part V, below, fits the *Concrete Machinery* model because the "first" step of that model is itself two-fold, as noted above. In any event, regardless of whether the procedure discussed in Part V precisely fits the *Concrete Machinery* model, I conclude, for reasons explained in Part V, that it is appropriate to consider an order of proceeding in this case that places issues of substantial similarity (in the evidentiary sense) ahead of issues of copyrightability.

#### V. Sharpening the Focus

In this case, if it is determined that neither a motion for summary judgment nor a motion for directed verdict should be sustained, how might issues involving substantial similarity (in the evidentiary sense) be submitted to the jury in a verdict form in the trial of this case?

To establish infringement, Lotus must prove, either directly or circumstantially, that Borland copied. Circumstantial proof of copying is made by showing access to the copyrighted work and substantial similarity (in the evidentiary sense). In the present case, I conclude that a reasonable trier of fact could find that the Quattro programs' user interfaces were not copied in their entirety from Lotus 1-2-3. However, because access is admitted, the record before me contains both circumstantial proof of copying (e.g., all of the commands in the Lotus 1-2-3 menu structure appear verbatim in the Quattro programs' menu structures in similar hierarchies) and direct proof of copying (e.g., Borland named certain files involved in the 1-2-3 emulation "123.rsc" and "123.mu"). It does not necessarily follow, however, that a jury must on this record find that some particular element was copied from Lotus 1-2-3 rather than being either independently created or copied from some other source.

For any elements of 1-2-3 that Lotus wishes to argue were copied, Lotus must demonstrate (either beyond dispute on motion for summary judgment, or else at trial by a prepon-

derance of the evidence) that the alleged copying from 1-2-3 occurred. Lotus may do so either directly or circumstantially. To demonstrate that the copying was illicit Lotus must also show substantial similarity in the mixed law-fact evaluative sense and copyrightability of those elements.

Returning to the question posed above (How might issues involving substantial similarity in the evidentiary sense be submitted to the jury?) I take note that one option to be considered is to begin with the less complex issue of copying and with a broad focus, then moving toward narrower questions concerning what elements of Quattro or Quattro Pro were copied from 1-2-3.

Would it be appropriate to frame the first question on a Verdict Form in a way such as this (with a similar question to follow for the other allegedly infringing work)?

#### Question 1

- (a) Do you find that the Quattro Pro user interface as a whole was copied from the Lotus 1-2-3 user interface as a whole?<sup>1</sup>

\_\_\_YES

\_\_\_NO

- (b) Do you find that the part of the Quattro Pro user interface called the "emulation interface" (also called the "1-2-3 compatible interface") as a whole was copied from the Lotus 1-2-3 user interface as a whole?

\_\_\_YES

\_\_\_NO

<sup>1</sup> It is not entirely clear that Lotus contends that part (a) of Question 1 should be answered YES, or that Borland contends that a NO answer to part (a) would be material if part (b) were answered YES. Perhaps the parties will agree, or one will persuade the court, that part (a) may be omitted.

- (c) If NO, do you find that some part, and, if so, which of the following part or parts of the Lotus 1-2-3 user interface were copied into some part of the Quattro Pro "emulation interface" (also called the "1-2-3 compatible interface")?

- |                          |                              |                             |
|--------------------------|------------------------------|-----------------------------|
| (1) The menu commands    | <input type="checkbox"/> YES | <input type="checkbox"/> NO |
| (2) The menu structure   | <input type="checkbox"/> YES | <input type="checkbox"/> NO |
| (3) The command sequence | <input type="checkbox"/> YES | <input type="checkbox"/> NO |
| (4) The long prompts     | <input type="checkbox"/> YES | <input type="checkbox"/> NO |
| (5) The macro facility   | <input type="checkbox"/> YES | <input type="checkbox"/> NO |
- 

To avoid any misunderstanding, I state explicitly that I have not determined that the foregoing draft of Question 1 is appropriate for use in this case. Nor have I determined what, if any, explanatory instructions should be given to the jury along with such a question. Even if a question of this general type is to be used, I will invite submissions by counsel of any different way of framing the question (and associated instructions) and why that different way would be more appropriate. Of course, counsel are invited to submit additional specific elements of alleged similarity about which there is dispute. Also, I do not suggest that all five elements on the present list ought to be included.

In particular, the term "macro facility" is not adequately defined in the parties' present submissions and would require definition. If "macro facility" means some element or characteristic of the user interface distinct from and not including the menu commands, menu structure, and command sequence, Question 1(c) might be appropriate as drafted. If, on the other hand, "macro facility" is to be defined in a broader sense that includes the menu command, menu structure, and command sequence elements, as well perhaps as the long prompts, but does not include other aspects of the 1-2-3 interface and for that reason is not the same as "the user interface as a whole,"

the foregoing draft of part (c) might be replaced by the following parts (c) and (d):

- (c) Do you find that the Lotus 1-2-3 macro facility as a whole was copied into some part of the Quattro Pro macro facility?

☐ YES ☐ NO

- (d) Do you find that some part, and, if so, which of the following part or parts of the Lotus 1-2-3 macro facility were copied into some part of the Quattro Pro macro facility?

- |                          |                              |                             |
|--------------------------|------------------------------|-----------------------------|
| (1) The menu commands    | <input type="checkbox"/> YES | <input type="checkbox"/> NO |
| (2) The menu structure   | <input type="checkbox"/> YES | <input type="checkbox"/> NO |
| (3) The command sequence | <input type="checkbox"/> YES | <input type="checkbox"/> NO |
| (4) The long prompts     | <input type="checkbox"/> YES | <input type="checkbox"/> NO |
- 

The purpose of including these alternative drafts in this Memorandum is to begin the process of focusing issues more precisely.

An advantage of beginning a Verdict Form with a question of this type on the subject of copying is that findings of the jury on the various parts of such a question would enable counsel and the court to focus more precisely on (a) the legal elements of "substantial similarity" (in the mixed law-fact evaluative sense) that are material after taking account of the jury findings, and (b) the legal (and factual, if any) elements of copyrightability that are material after taking account of the jury findings. Surely some of the legal and factual issues that cannot be dismissed as immaterial before taking account of the jury findings would be mooted by the findings. For example, if a jury finds that the emulation interface as a whole was copied from the 1-2-3 interface as a whole, copyrightability questions with respect to less of the user interface

than the whole might be moot. On the other hand, if the jury finds that neither the emulation interface nor certain individual elements of it were copied from Lotus 1-2-3, but certain other elements were copied, the material copyrightability questions would be limited, others having been mooted.

I have proposed presenting the question to the jury as one of "copying" rather than "substantial similarity" even though access is undisputed. I have done so because a mere finding of "substantial similarity" in the circumstantial evidentiary sense may be inconclusive on the issue of copying. That is, even if "substantial similarity" in the evidentiary sense creates a presumption of copying, it may be that the presumption is rebuttable and that the jury should be so instructed. If so, Borland might still persuade the jury that the circumstantial inference of copying has been rebutted by extrinsic evidence offered by Borland. I do not at this time decide any of the questions of law suggested in this paragraph.

Consideration may be given to trying Phase One on copying and commencing Phase Two of trial immediately after Phase One, and before the same jury, to address any remaining questions that could properly be submitted to the jury on issues of "substantial similarity" in the mixed law-fact evaluative sense and "copyrightability."

#### VI. *Framing the Issues of Copyrightability*

A good place to begin the exploration of potential options in this case regarding submission to a jury of issues of copyrightability (if such issues are to be submitted to a jury) is to consider whether it would be appropriate to submit a single copyrightability mixed law-fact question, framed in terms of the applicable legal standard. Before I suggest such a jury question, I first state the standard for deciding copyrightability that will be used in this case, absent further guidance from higher authority before the date of trial.

#### A. The Standard for Deciding Copyrightability

Having fully considered the submissions now before the court, I conclude that, for reasons stated in the *Paperback Opinion*, 740 F.Supp. at 54-62, and supplemented here, the copyrightability issues in this case should be determined by the legal standard used there and summarized as follows:

FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, *along the scale from the most generalized conception to the most particularized*, and choose some formulation—some conception or definition of the "idea"—for the purpose of distinguishing between the idea and its expression.

\* \* \*

SECOND, the decisionmaker must focus upon whether an alleged expression of the idea is limited to elements essential to expression of *that* idea (or is one of only a few ways of expressing the idea) or instead includes identifiable elements of expression not essential to every expression of that idea.

THIRD, having identified elements of expression not essential to every expression of the idea, the decisionmaker must focus on whether those elements are a substantial part of the allegedly copyrightable "work."

*Id.* at 60-61.

In reaching this conclusion, I have fully considered *amicus curiae* briefs filed in this case, with leave of court. Compared with the great majority of *amicus curiae* briefs filed in courts, two of the *amicus* submissions in this case are distinctive: they do not purport to advise the court how the present case should be decided, and they are not filed on behalf of clients who have a special interest aligned with that of one or the other of the parties to the case.<sup>2</sup> The *amicus* submission of the

<sup>2</sup> A third *amicus* brief filed by the Software Entrepreneur's Forum does take the position that Quattro and Quattro Pro ought to be determined not to infringe Lotus' copyright. However, that brief suggests find-

Register of Copyrights concerns issues that I do not reach in this Memorandum, and I do not discuss it further.

The second *amicus* submission was filed by eleven professors of law, teachers of courses in the intellectual property field, who express a concern that the Opinion in *Lotus v. Paperback* and the standard of decision developed in that Opinion, quoted above, give inadequate attention and emphasis to the distinction between a copyrightable expression and a useful process. Although this distinction is noted in the *Lotus v. Paperback* Opinion—e.g., 740 F.Supp. at 53-58—I accept the point that no explicit reference to “process” appears in the standard of decision quoted above. Because I recognize the value of reminding decisionmakers (whether judges or jurors, a point to which I return later in this Memorandum) of the distinction between a useful process and an original expression, I will add a reference to that distinction, restating the standard in the following way:

FIRST, in making the determination of “copyrightability,” the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation—some conception or definition of the “idea,” “system,” “process,” “procedure,” or “method”—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

\* \* \*

SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essen-

ings that ought to be made rather than the legal rule that must be applied. Because the court does not make findings on motions for summary judgment, and even as to other matters addressed in this Memorandum I am not making findings, I do not address that brief further.

tial to every expression of that idea, system, process, procedure, or method.

THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable “work.”

*Id.* at 60-61 (added words emphasized, other emphasis deleted). If the answer to the THIRD step is YES, then the expressive elements of the work, taken together, are copyrightable. Copyright protection extends, of course, only to the expressive elements—not to anything more. To demonstrate entitlement to relief, Lotus will be required to prove that Borland copied expressive elements (that is, the particular form of expression and not just the methodology, process, or idea of the user interface of Lotus 1-2-3) and that as a result there is a substantial similarity (in the mixed law-fact evaluative sense) between expressive elements of 1-2-3 and an allegedly infringing Borland program.

I need not and do not tarry over whether this refinement of the stated standard of decision simply makes explicit something that was implicit in *Lotus v. Paperback* or instead is a modification of the standard stated there. In any event, laboring as I am in territory that is uncharted, I conclude that the sources of authority I am bound to respect—constitutional, statutory, and decisional—leave me the choice and perhaps even the responsibility to make this refinement and to do so explicitly.

Apparently, if not explicitly, a premise of Borland’s argument is that once some aspect of a “computer program” (as that phrase is used in the statute, 17 U.S.C. § 101) is determined to be a “process” or “system” (as those words are used in the statute, 17 U.S.C. § 102(b)), no part of that functional aspect is copyrightable. (It is not entirely clear whether the argument is that anything “functional” is not “expressive,” or instead that even “expressive” elements of anything that is “functional” are not “copyrightable.” The choice between

these two different ways of phrasing the argument seems, in any event, more linguistic than substantive.) Once functionality is demonstrated, Borland argues, the inquiry about copyrightability ends, and no copyright is possible. This premise of Borland's position is flawed.

The Professors' Amicus Brief might be interpreted as implicitly if not explicitly supporting Borland's premise. In any event, Borland broadly asserts that "all systems are uncopyrightable." Docket No. 141, at 94. Of course, if Borland means only that being a "system" does not make something copyrightable, the point is plainly meritorious but does not help one decide a case such as this one. But Borland appears to be asserting more—that if some part of a computer program is a "system, process, procedure, or method," 17 U.S.C. § 102(b), no copyright of any aspect of that part of the program is possible. If, in so arguing, Borland means that the fact that a computer program is a "system" precludes copyrightability of every part and aspect of that program (or even that if some part of a program is a "system," copyrightability is precluded as to all aspects of that part) the argument is deeply flawed. See *Kieselstein-Cord v. Accessories By Pearl, Inc.*, 632 F.2d 989 (2d Cir.1980) (holding that expressive elements of utilitarian belt buckles were copyrightable). The fallacy becomes apparent as we examine more closely the proposed bright-line rule that "all systems are uncopyrightable." Even source code and object code would not be copyrightable if this rule were the law.

It is worth noting that in this discussion I do not depend upon judicially or academically developed conclusions about fundamental truths, even though the basic nature of copyright law is relevant. Rather, the central point is that because courts are bound by the congressional mandate that something in computer programs is copyrightable, I must reject Borland's premise. Of course, if Congress should at some future time determine that the balance it has struck with respect to the copyrightability of computer programs is not in the best interests of the programming industry and the public, Congress (but not this court) is free to change that balance.

In discussing this issue, as shorthand for convenience only (and with readiness to reconsider should it be suggested that the shorthand has different substantive implications from the longer phrase), I will use the term "process" to refer to "system, process, procedure, or method" as that phrase is used in the statute.

"Process," like "idea," is an abstraction—a creature of the human intellect. A machine may be constructed of pieces of metal, wood plastic, and other materials, put together with nuts, bolts, bearings, and adhesives. A process, on the other hand, is not composed of materials. Instead it is a set of ideas about how to do something. The fact that it is an idea about doing does not make it any less an idea.

Patent law establishes legal rules for process patents that are different in some respects, and not in others, from the rules applying to other patents. The mere fact that patent law allows a means of legal protection for a process, however, does not establish that there is not also some protection in copyright law. Certainly the area of legal protection under the separate legal regimes is not co-extensive, but it is equally clear that there is no particular reason to believe there should never be any area of overlap. Indeed, precedent recognizes some overlap. *E.g.*, *Mazer v. Stein*, 347 U.S. 201, 217, 74 S.Ct. 460, 470, 98 L.Ed. 630 (1954) ("We do hold that . . . patentability . . . does not bar copyright. . . ."). Thus, the fact that a particular computer program may be protected to some extent under one of these legal regimes does not mean it cannot be protected to any extent under the other.

Thinking about copyright protection for computer programs may begin with the point that a person who conceives a "process" shows or tells—uses some form of communication—to make the "process" accessible to another. Showing and telling are modes of expression.

Thus, there is a closely analogous if not precisely identical dilemma about distinguishing an idea from expressions of the idea and a "process" from expressions of the "process." I need not, and do not, repeat here the explanation of that dilemma and of statutory and decisional markers that led me in *Lotus*

v. *Paperback* to the formulation of a standard to be applied to the facts of a particular case to decide whether an aspect of a computer program is a copyrightable expression.

In a broad sense of the term "process" (or the longer phrase for which it stands here), every aspect of a "computer program" is part of a process. Nevertheless, we have a statutory mandate that some aspects of a "computer program" may be copyrightable. Borland and virtually all others who discuss the matter recognize that "source code" and "object code" have copyrightable elements. Yet each "code" is surely a part of a "system, process, procedure, or method." Imagine the response you would receive from a good programmer if you told her that her "source code," or "object code," or both, lacked "system" and "method" and, either taken together or separately, simply could not be regarded as an effective "process" or "procedure" for communicating to the computer what it should do!

Bowing to the congressional mandate and to widely expressed views about the copyrightability of code, Borland—after full explication of a proposed interpretation of *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879), and its progeny—continues with the point that the "long-standing copyright principles" applicable to novels, poems, and other traditional literary works have been applied also to computer programs to provide significant protection to software developers.

For example, under those long-standing principles, *the sequence, structure and organization of the program's code*, in addition to the text of the code itself, *may be protected by copyright*—under the same rationale that protects the detailed plot line and structure of a play in addition to the play's actual dialog.

Docket No. 141 (emphasis added). Borland also concedes, as it must, that this protection of the sequence, structure, and organization may extend to the user interface as well as source code and object code.

Borland's flawed argument (for a bright-line rule that a finding of "process" defeats "copyrightability," regardless of the originality of any particular expression that enables a person other than the creator to use it) must be rejected because it is fundamentally inconsistent with the congressional balance struck in the Copyright Act. The argument is in its nature a one-conclusive-element argument. Expressed in the form of if-then logical operations of the sort a computer might execute upon pairs of binary values, such an argument asks the court to conclude (a) "if process, then not copyrightable," (b) "if idea, then not copyrightable," and (c) "if patent protection for process, then no copyright protection for process." Arguments of this type, urging courts to adopt an overriding rule that one element of a total set of interwoven circumstances be declared the only legally relevant element are like discredited arguments that courts should select one among all the antecedents of an event in human affairs and declare it to be *the proximate cause*. See W. Page Keeton et al., *Prosser & Keeton on the Law of Torts* 266 (5th ed. 1984). These are arguments that one aspect of the total circumstances be treated in law as overriding and conclusive. Arguments of this type are often encountered in the legal system, but far less often sustained. Common sense tells us that life and life's experiences are not so neatly separated into discrete categories, without overlap. The legal system aims at reasonable accommodations among conflicting legitimate interests and rarely holds that one such interest totally overrides all others. Moreover, in the present context, respect for the congressional mandates in the Copyright Act requires that Borland's proposed bright-line rule be rejected.

Borland's argument that *Baker v. Selden* adopted a bright-line rule that governs this case is a gross overstatement not only of what the Supreme Court *did* in that case but as well of what the Court *said*. The Court did not there face the issues presented in the present case. Indeed, the Court did not even address issues of expression within the plot, sequence, organization, or structure of artistic works—issues that were

addressed by Learned Hand in his opinions from which I took guidance in deciding *Paperback*.

Attributing to the Supreme Court one's own extrapolations beyond what the Court did and said is a form of reasoning a lower court should not adopt. It is wiser and more in keeping with the judicial role to recognize candidly that we, not the Justices of the Court, are doing the extrapolating. In hearing and deciding *Baker v. Selden* in 1879 the Court did not, and indeed could not, foresee computer programs. The Court was not speaking about computer programs or even about the kinds of issues, more readily inferred to be foreseeable in 1879, that were explicitly addressed in Learned Hand's opinions a half-century later and beyond. Attributing to the Court's decision in 1879 conclusions that counsel or courts may draw today from *Baker v. Selden* by extrapolation is being less than fully candid about the extent to which decisions must be and are being made today. Nor can we attribute to Congress, in its much more recent action, a manifestation of intent drawn from a supposed legislative history arguably appearing in a CONTU report to Congress. Congress speaks primarily by what *Congress says* formally by its enactment. Even though legislative history regarding expressions by members of Congress during the legislative process is sometimes used by courts in aid of determining the meaning of the formal enactments of Congress, there is no support for treating as legislative history what another person or entity says to Congress. Thus, to be candid about what we are doing and faithful to congressional as well as precedential guidance, we cannot say that *Baker v. Selden*, or the CONTU Commission's interpretation of that decision, established a bright-line rule, "if functional, not copyrightable." As with "causation," we must delve more deeply to understand such concepts as "process," "expression," "copyrightability," and "substantial similarity."

#### B. Drafting a Jury Interrogatory to Test Its Viability

In deciding copyrightability issues in the present case I will apply the standard formulated in *Lotus v. Paperback*, with the

refinement described above to take account of the concern appropriately expressed in the Professors' Amicus Brief. If "copyrightability" is to be submitted to the jury, then Question 2 of a Verdict Form in which Question 1 concerned copying might read as follows:

#### Question 2

In answering this question, you are to apply the following legal standard:

FIRST, in making the determination of "copyrightability" of the user interface of 1-2-3 as a whole, or of some element or elements of that interface, focus upon alternatives that counsel may suggest, or that you may conceive, along the scale from the most generalized conception of that user interface, or some element or elements of that interface, to the most particularized, and choose some formulation—some conception or definition of the "idea," "system," "process," "procedure," or "method" of the interface, element, or elements—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

SECOND, focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work." If they are, then you will find those expressive elements, and no more, copyrightable.

Applying this standard, do you find by a preponderance of the evidence that the user interface of Lotus 1-2-3 contained copyrightable elements?

\_\_\_ YES \_\_\_ NO

As one considers this option, it becomes immediately apparent that its effect is to give the jury virtually uncontrolled discretion. Prohibitions against probing into jury deliberations—e.g., Fed.R.Evid. 606(b)—would preclude courts from determining whether a jury understood and correctly applied the court's explanation of the standard in its charge to the jury. The only judicial control against improper verdicts would be in decisions whether or not to direct a verdict and whether or not to grant a motion for judgment notwithstanding the verdict.

Might counsel and the court devise some greater measure of assurance that the jury understands and faithfully applies congressional directives regarding the scope of copyright protection for computer software programs? Would it be a step in that direction to use the foregoing formulation as part (a) of a question, with something like the following as part (b)?

(b) If YES, is each of the following a part of the user interface in which you find expressive elements, which you also find to be at least part of the basis for your answering YES to Question 2(a)?

- |   |         |        |
|---|---------|--------|
| (1) The menu commands   | ___ YES | ___ NO |
| (2) The menu structure  | ___ YES | ___ NO |
| (3) The command sequence                                      | ___ YES | ___ NO |
| (4) The long prompts  | ___ YES | ___ NO |
| (5) The macro facility  | ___ YES | ___ NO |
| (6) The overall appearance of some part of the user interface | ___ YES | ___ NO |

Answers to the subparts of a question of this kind might at least enable a court, after verdict, to give somewhat more reasoned consideration to the relevance and validity of an argument—such as is advanced by Borland here—that, as precedent, *Lotus v. Paperback* should be limited by the finding that Paperback copied the “user interface as a whole,” and that the present case is materially different because here, even if the facts are construed most favorably to Lotus, Quattro and Quattro Pro are substantially similar to less of the user interface of Lotus 1-2-3 than was VP Planner. (I do not now determine the validity of this argument.)

Even if an addition such as part (b) might be useful in some circumstances, however, it would not eliminate, or even substantially mitigate, another serious problem to which I turn next.

#### VII. Should Any Copyrightability Issues Be Submitted to a Jury?

Is a fact question bearing upon copyrightability to be submitted to a jury if a timely demand for jury trial has been filed and the fact is genuinely in dispute—that is, reasonable persons might differ about the answer to the question?

Though debated often, as yet this question has not been explicitly answered either by statute or by precedents. Cases that arguably bear upon the question can at the least be distinguished on the facts and thus do not speak directly to the precise kinds of copyrightability issues that are presented in this case. E.g., *Kregos v. Associated Press*, 937 F.2d 700 (2d Cir. 1991) (reversing summary judgment on issue of “creativity”). *Kregos* did not involve the distinctive entanglement problems that are presented in computer-program copyrightability cases.

In *Lotus v. Paperback*, even though the question as to whether copyrightability might be a jury issue had been discussed extensively in pretrial conferences, the court was not required to and did not decide it. The parties removed it from contention by entering into a stipulation. As part of an agree-

ment for a phased trial in which Phase One was to be before the court without a jury, the parties to that case stipulated that any fact question relevant to copyrightability would be decided by the court. No such stipulation has been made in this case, and I now address the issue.

#### A. Role of Judge, Jury, and Witnesses

The legal test for determining copyrightability both as formulated in *Lotus v. Paperback* and as refined in Part VI, above, is a standard requiring an evaluative mixed law-fact determination, as distinguished from a bright-line rule calling for a finding about disputed historical facts such as who did what, where, and when. Moreover, this standard is far more heavily loaded with public policy implications than most other standards more commonly used in law, of which the negligence standard is an example. Juries applying the copyrightability standard would not be required or even permitted to explain their reasoning. They would be free as a practical matter to reach decisions inconsistent with the balance struck by Congress, as interpreted by the courts. Inconsistencies among verdicts could be expected to introduce a lawless element into the administration of justice in copyright cases, quite inconsistent with the aim of treating like cases alike.

In the circumstances of a particular case, the answer to Question 2 (or any alternatives to Question 2 we might envision), is an essential premise of a reasoned application of the accommodation of conflicting policy interests reached by Congress in determining that computer software programs are copyrightable. For this reason, I conclude that the application of the standard to a particular case is a ruling more closely analogous to traditional judicial lawmaking to fill the interstices of statutes than to traditional factfinding. It is appropriately treated as a ruling of law. Thus, even if a court treats the answer to Question 2 as a "factfinding," the court may conclude also that the answer is a finding of a premise fact—a finding of a fact that serves solely as a premise for a ruling of law. See Robert E. Keeton, *Legislative Facts and Similar Things: Deciding Disputed Premise Facts*, 73 Minn. L.Rev. 1

(1988). Like a finding about whether transporting gasoline on a public highway, or blasting with dynamite, or cropdusting, or building an earthen dam is an abnormally dangerous activity, *id.* at 19-20, 57-69, perhaps, this is a finding to be made not by juries, case by case, but by a lawmaker (Congress, to the extent that it can do so in broadly applicable statutory guidelines, and courts to the extent necessary to fill out details essential to application of those broad guidelines in particular types of cases).

The position that this kind of finding is a premise factfinding is supported also by the certainty that even if patterns of jury verdicts might develop over time and become sufficiently predictable to be reasonably described as part of the "law" of copyright, this element of the law would be developed undercover rather than, as is expected of lawmaking through judicial precedents, by "reasoned decisionmaking, candidly explained." Robert E. Keeton, *Judging* 1-2 (1990). Moreover, even if patterns were to develop sufficiently to enable lawyers to predict outcomes for clients with some confidence, still the patterns could not be employed to protect an individual litigant, plaintiff or defendant, against the harsh consequences of a deviant verdict.

The length and complexity of a jury trial of issues of copyrightability would be affected also by disputes regarding admissibility of opinion evidence. The affidavits submitted in this case by the parties, in support of their respective positions on the cross-motions for summary judgment, present what is in essence a clash of policy arguments by experts. Moreover, an impartial factfinder may reasonably infer that the policy positions the experts advance correlate better with their respective views about what the law of copyrightability should be than with considered opinions about the policy accommodation Congress has struck and courts and juries are bound by oath to respect. This point is supported by the following passage from Borland's Memorandum (Docket No. 141):

More importantly, Borland's witnesses include distinguished industry executives who have years and years

of actual experience in bringing software products to market. Galler [a Lotus witness], in contrast, can only theorize and speculate about such issues. Unlike Galler, Borland's experts can authoritatively discuss the consequences to software development of the untoward extension of copyright law Lotus seeks in this case.

*Id.* at 28. Thus, the affidavits of experts submitted to this court for consideration in ruling on the motions for summary judgment are aimed more at persuading this court to a view of the law than to the existence or nonexistence of a genuine dispute of fact.

If copyrightability is held to be an issue for jury determination, courts and counsel must work out answers to extraordinarily difficult issues regarding admissibility of opinion evidence.

No provision of the Copyright Act declares explicitly that issues of fact bearing on copyrightability shall be submitted to juries, or instead shall be decided nonjury. Nor does the Constitution, including the Seventh Amendment. Thus, the answer to the question who shall decide such issues must necessarily be fashioned by courts. Courts are not free, however, to fashion whatever answers they may deem best. They must, instead, seek answers consistent not only with the explicit constitutional and statutory mandates but also with the implications of those mandates for other issues not explicitly addressed. A determination that issues of copyrightability are to be resolved by juries would have such severe adverse effects on the aim of assuring that like cases are treated alike and on the complexity and cost of litigation in computer software copyright cases that in practical effect the scope of copyright protection congress manifestly intended could not be achieved. The practical certainty of many outcomes inconsistent with the congressional accommodation among highly valued but conflicting interests, manifested in the Copyright Act, weighs heavily in favor of the conclusion that treating copyrightability issues as exclusively for courts, not juries, at least in computer software cases, is the decision more compatible with the congressional mandate. So also does the

object of avoiding the practical certainty of increased length, complexity, and cost of litigation.

For all these reasons, I reach the tentative conclusion (which I will be prepared to reconsider as explained in Section B, below) that at least in the circumstances of this case (and probably more generally, though I need not so determine here), the issue or issues of copyrightability, including any fact questions bearing upon them, must be determined by the court, not the jury.

#### B. Reconsideration

Part II, above, alludes to the value of thinking seriously about the framing of jury questions and instructions well in advance of trial because the attempt to do so helps sharpen the focus on issues that are best considered early in the history of the case. Part VII-A tentatively determines that copyrightability issues in this case are to be determined by the court, not the jury. Before so ruling finally, however, I will allow each party an opportunity to submit a proposed draft of any question bearing on copyrightability that it contends, even in the alternative, is a question that should be submitted to the jury. If the draft question uses legal jargon rather than plain English only, the draft must be accompanied by proposed instructions to the jury explaining the jargon. Submissions and responses may be filed on the schedule set in the Order below.

#### C. Ambiguity of the Parties' Contention Regarding Copyrightability

It is not now apparent which individual elements of 1-2-3 the plaintiff (Lotus) alleges to be copyrightable, nor has either party made clear its contention about what aspects of various elements of the 1-2-3 user interface are or are not expressive aspects.

Lotus' Statement of Material Facts as to Which There is No Genuine Issue Pursuant to Local Rule 56.1 (Docket No. 149) includes the following paragraphs:

32. The selection of which, and how many, commands to place in each menu level, and the organization of the successive menu levels into a coherent and intuitive menu structure, was an important and creative consideration in the development of 1-2-3. 1-2-3's creators, particularly Mr. Kapor, devoted substantial effort during the late stages of the program's development to selecting the words and structural organization for 1-2-3's menus. (Kapor Aff., ¶¶ 72-101.)

33. Thus, the 1-2-3 menu commands, their organization and sequence, the 1-2-3 menu tree, and the overall user interface reflect an original expression of a spreadsheet program. This expression is not dictated by functional constraints and contains a significant degree of communicative content to the user. (First Galler Dec., ¶ 207.)

*Id.* at ¶¶ 32, 33.

Borland responds that paragraph 32 is uncontested and immaterial and that paragraph 33 is contested or non-factual. Borland adds, *inter alia*, that "[t]he suggestion in the first sentence of No. 33 that the 1-2-3 commands and command hierarchy represent copyrightable 'expression' is not a statement of fact but rather constitutes an asserted conclusion of law," which Borland disputes. Even though contending that this is an issue of law, Borland has nevertheless resisted trying it separately and first. Thus, I understand Borland's position to be that if the court does not decide this issue for Borland as a matter of law, Borland wishes not to be foreclosed from contending, in the alternative, that copyrightability in general, and this issue in particular, should be submitted to the jury.

I conclude that a decisionmaker (judge or jury) cannot determine the answer to the dispute over Lotus' ¶ 33 without making an evaluative application, to this case, of the standard formulated in Part VI of this Memorandum. One might reasonably argue that an issue such as this, if one for determination by the court, can properly be decided on motion (or

cross-motions) for summary judgment, even if the issue is one on which reasonable persons may differ in the circumstances of this case. Nevertheless, even without so deciding now, I conclude that on the present submissions I am left with a sense that this is a genuinely disputed and debatable issue. Unless one of the parties can persuade me otherwise in further submissions permitted by the attached Order, I will leave this issue to be decided on a full record developed in a first phase of trial, and with the benefit of a sharper focus than present submissions accomplish as to precisely what issues of fact and law should be decisive of this case.

In particular, the dispute between the parties regarding paragraph 33 is one I cannot decide on the present submissions. If the menu commands or menu command structure were dictated solely by functional concerns, then those elements may not be copyrightable. *See Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1145 (2d Cir.1987) ("conceptual separability" test examines whether expression exercised independent of functional influences). But the mere fact that functional concerns were influential does not establish that no copyrightable expression appears in the menu commands, the command structure, or elsewhere in the user interface.

#### VIII. Substantial Similarity Revisited (Illicit Copying)

It is not enough that an allegedly infringing work contain matter copied from a copyrighted work. Lotus must demonstrate that similarities between the works are substantial. However, not all similarities may be taken into account. For example, the fact that both programs are spreadsheet programs renders them substantially similar in many respects. Nevertheless, such similarities may form no part of a conclusion that the works are substantially similar in the mixed law-fact evaluative sense—that is, for purposes of assessing *illicit copying*—because the idea (or process) of a spreadsheet program is not copyrightable. It is naturally to plaintiff's

advantage to place as many copyrightable elements before the court and jury on the issue of substantial similarity as possible not only because the sum increases with each added element but also because the whole may be more than the sum of its parts. Therefore, a determination as to which elements of Lotus 1-2-3 are copyrightable, alone or together, is interwoven with the issue of substantial similarity.

On the other hand, as I have concluded above, it is by no means apparent that all copyrightable elements of 1-2-3 have been copied in the Quattro programs. Lotus, presenting as its primary contention that *Paperback* is controlling, has not formulated for the court or for Borland its precise contentions (in the alternative to its contention that the user interface as a whole was copyrightable and was copied) as to which elements of 1-2-3, separately or in combination, were copyrightable and were copied.

Borland, on the other hand, at points in its submissions, appears to contend that it is entitled to summary judgment both on the issue of substantial similarity and on the issue of copyrightability because if anything at all in a computer program's user interface is copyrightable it is only the user interface as a whole, and the elements of the Quattro and Quattro Pro user interfaces that are like elements of the Lotus 1-2-3 interface are only a very small part of the *Quattro and Quattro Pro user interfaces*. This is an argument not likely to prevail in any court. Copyrightability is determined by the characteristics of the allegedly protected work, not by the characteristics of the allegedly infringing work. A copier who incorporates a copyrighted short story of twenty pages length into a multi-volume set of books will not get far with the contention, in litigation between the copyright owner of the short story and the copier, that the short story is not copyrightable because it is such a small part of the multi-volume set. Nor will the copier win on the substantial similarity issue by showing that although the entire copyrighted work was copied it constituted only a very small part of the overall set of books produced by the copier.

I need not and do not rule at this time on the appropriate framing of the issues that must be decided in this case about the legal significance that may attach to findings of fact about the disputed scope of similarities between the user interfaces of the different works. To say the least, however, Borland is not entitled to a summary judgment on this extraordinary ground it appears to be asserting.

If Borland responds that I have misinterpreted its submissions and it did not mean to assert such an extreme position, its future submissions regarding precisely what questions it proposes to submit to the jury and what expert opinions it proposes to proffer may clarify its contentions. As to opinion testimony, it seems likely indeed the court should preclude the expression in the presence of a jury of an opinion that "substantial similarity" does not exist when the elements of the user interfaces that are alike constitute only a small percentage of the allegedly infringing product. Whether that is correct or not is not a question of fact on which expert opinion may properly be placed before a jury.

The argument that a little copying—beyond "fair use," 17 U.S.C. § 107—here and there in a very large work should be allowed to promote the development of great works is a policy argument Congress has rejected. It should not be heard by a jury. The issue is one of law as to which the court will receive arguments of counsel, outside the presence of the jury, both to determine the scope of admissible opinion testimony and to determine the framing of questions of fact to be submitted to the jury.

Neither, on the other hand, will the jury hear the argument that copying just the menu terms, or just the menu structure, or just a command sequence, is enough to establish infringement. Whether this is so or not is a question of law to be argued to the court, not the jury. To the extent that strictly factual answers from the jury may be helpful to the court and counsel to focus the legal issues to be argued, however, questions may be framed to elicit those answers.

## ORDER

For the foregoing reasons, it is ORDERED:

(1) The cross-motions for summary judgment (Docket Nos. 30 and 87) are denied.

(2) Renewed motions for summary judgment may be filed by April 10, 1992 and responses by April 20, 1992.

(3) Submissions of both parties pursuant to Part VI of the foregoing Memorandum shall be filed on the schedule in (2).

(4) The several motions regarding the submission of confidential materials (Docket Nos. 140, 156, and 160) are dismissed as moot.

A conference to consider all pending matters and to set a trial date is scheduled.

UNITED STATES DISTRICT COURT  
D. MASSACHUSETTS

Civ. A. No. 87-76-K

June 28, 1990

---

LOTUS DEVELOPMENT CORPORATION,

*Plaintiff,*

—v.—

PAPERBACK SOFTWARE INTERNATIONAL  
and Stephenson Software, Limited,

*Defendants.*

---

Henry B. Gutman, Kerry L. Konrad, John S. Beckerman, Karen F. Conway, O'Sullivan Graev & Karabell, New York City, and Thomas J. Dougherty, Skadden, Arps, Slate, Meagher & Flom, Boston, Mass., *for Lotus Development Corp.*

Lawrence G. Papale, Cannata, Genovese & Papale, San Francisco, Cal., and Edward C. Saltzberg, Warner & Stackpole, Boston, Mass., *for Stephenson Software, Ltd.*

Paul R. Gupta, David A. Guberman, Barbara O'Donnell, Brian C. Levey, Nereyda F. Garcia, Kenneth R. Berman, Sherin and Lodgen, Boston, Mass., and Walter G. Murphy, Peter C. Kober, Murphy, DeMarco & O'Neill, P.C., Boston, Mass., *for Paperback Software Intern.*

---

## OPINION

KEETON, District Judge.

The expression of an idea is copyrightable. The idea itself is not. When applying these two settled rules of law, how can a decisionmaker distinguish between an idea and its expression?

Answering this riddle is the first step—but only the first—toward disposition of this case in which the court must decide, among other issues, (1) whether and to what extent plaintiff's computer spreadsheet program, Lotus 1-2-3, is copyrightable, (2) whether defendants' VP-Planner was, on undisputable facts, an infringing work containing elements substantially similar to copyrightable elements of 1-2-3, and (3) whether defendants' proffered jurisdictional and equitable defenses are meritorious.

Phase One of this case was tried to the bench. By agreement of the parties, in Phase One, the court shall

resolve all legal and factual issues concerning the liability, if any, of defendants . . . for the claims of copyright infringement brought by plaintiff . . . and all defenses thereto, including but not limited to all factual and legal issues concerning the copyrightability of Lotus' Works [1-2-3, releases 1.0, 1A, and 2.0], and *excluding only*: (1) issues of fact, if any, requiring jury determination concerning defendants' alleged copying of any protected expression from Lotus' Works in Defendants' Works [VP-Planner and VP-Planner Plus]; [and] (2) factual issues concerning defendants' possible copying of the source or object code for Lotus' Works. . . .

Stipulation and Order Regulating Phased Trial, § I(A) (Docket No. 246).<sup>1</sup>

<sup>1</sup> This case was originally consolidated with *Lotus Development Corporation v. Mosaic Software*, Civil Action No. 87-74-K, which involves Mosaic's computer spreadsheet program The Twin. Although Phase One of this case was intended to resolve certain additional legal and factual issues concerning the liability of Mosaic Software for plain-

This Opinion sets forth findings of fact and conclusions of law that are central to deciding this controversy. Fed.R.Civ.P. 52(a). The court adopts, as additional findings, all proposed findings to which no party objected. See Docket Nos. 250, 251.

The outcome of this case depends on how this court, and higher courts on appeal, should answer a central question about the scope of copyrightability of computer programs. For the reasons explained in this Opinion, I conclude that this question must be resolved in favor of the plaintiff, Lotus.

# I. A BACKGROUND STATEMENT ABOUT COMPUTERS, COMPUTER PROGRAMS, AND COPYRIGHTABILITY

Though their influence in our society is already pervasive, digital computers—along with computer “programs” and “user interfaces”—are relatively new to the market, and newer still to litigation over “works” protected by intellectual property law.

Digital computers (hereinafter referred to as “computers”) are machines currently used to perform three types of functions electronically: (1) arithmetic calculations; (2) logical operations (*e.g.*, comparing values to determine whether one is larger); and (3) storage and display of the results. Because computers can perform millions of operations of these types in a single second, they can be used to solve problems too complex, or too repetitious and boring, to be solved manually. Developments to the current state of the art have already transformed many areas of business, educational, and recreational activity, and they support speculations about more striking achievements in the future.

A personal computer system consists of hardware and software. The hardware includes the central processing unit (“CPU”), which contains the electronic circuits that control the computer and perform the arithmetic and logical func-

tiff's claims of copyright infringement, that case was severed on the second day of this first phase of trial for reasons not here relevant. See Docket No. 174 (February 8, 1990).

tions, the internal memory of the computer ("random access memory," or "RAM"), input devices such as a keyboard and mouse, output devices such as a display screen and printer, and storage devices such as hard and floppy disk drives. The software includes one or more computer programs, usually stored magnetically on hard or floppy disks, along with such items as instruction manuals and "templates," which are pieces of plastic that fit around the function keys on the keyboard, identifying the specific functions or commands that can be invoked by those keys. A personal computer system can also include "firmware," or "microcode":

Microcode is a set of encoded instructions . . . that controls the fine details of the execution of one or more primitive functions of a computer. Microcode serves as a substitute for certain elements of the hardware circuitry that had previously controlled that function.

Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 Duke L.J. 663, 677.

Computer programs are, in general, divided into two types: operating system programs and application programs. Operating system programs—such as DOS, XE-NIX, and OS/2—are programs that control the basic functions of the computer hardware, such as the efficient utilization of memory and the starting and stopping of application programs. Application programs are programs that permit a user to perform some particular task such as word processing, database management, or spreadsheet calculations, or that permit a user to play video games.

This case concerns two competing application programs—Lotus 1-2-3 and VP-Planner—which are primarily spreadsheet programs, but which also support other tasks such as limited database management and graphics creation. Programs such as these, because they can perform several different kinds of tasks, are called "integrated" application programs.

Congress has defined "computer program" as follows:

A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

17 U.S.C. § 101 (1988). This "set of statements or instructions," in its literal or written manifestation, may be in the form of object code or source code. It may also be represented, in a partially literal manifestation, by a flowchart. A copyrightable work designed for use on a computer may include, as well, text that appears, for example, in a problem manual or a manual of instructions. These elements of text, however, ordinarily are not referred to in the industry as part of a "computer program" unless they appear on the computer screen and serve a purpose like that of the components of a "help screen" available to a user whenever needed. Elements of this textual type are not at issue in this phase of this case.

Computer programs are typically written in some form of computer programming "language." The "lowest"-level computer programming language is machine language, which is a binary language written in "bits" (*B*inary *di*giT*S*). Each bit is equal to one binary decision—that is, to the designation of one of two possible and equally likely values, such as an "on"-"off" or "yes"—"no" choice. These binary decisions, the only kind that a typical computer can understand directly, are commonly represented by 0's and 1's. A sequence of eight bits (which allows 256 unique combinations of bits) is commonly called a "byte" ("by eight"), and 1024 bytes form a "kilobyte" (commonly referred to as "K," *e.g.*, sixty-four kilobytes is "64K"). Machine language may also be represented in hexadecimal form, rather than in binary form, by the characters 0-9 and A-F, where "A" represents 10, "B" represents 11, and so on through "F," which represents "15." In hexadecimal machine language, only two rather than eight characters are required to allow for 256 unique combinations (*e.g.*, 37 instead of 00110111, each of which represents the 55th of 256 combinations; 7B instead of 01111011, each of which represents the 123rd of 256 combinations; EA instead

of 11101010, each of which represents the 234th of 256 combinations). The computer is able to translate these hexadecimal instructions into binary form. Other versions of machine language are represented in decimal (0-9) and octal (0-7) form.

An object program, or object code, is a program written in machine language that can be executed directly by the computer's CPU without need for translation. For example, in the machine language of a certain computer, the instructions to divide the value in "B" by the value in "C" and add that number to the value in "A" may be represented by the following sequence of instructions (in binary form):

```
0010000000010001; 1000000011010010;
1101000000010000.
```

An "intermediate"-level programming language is assembly language. Rather than in bits, assembly code is written in simple symbolic names, or alphanumeric symbols, more easily understandable by human programmers. For example, the calculation described above may be represented, in the assembly language of a certain computer, as follows:

```
LOAD B; DIV C; ADD A.
```

Because of the primitive nature of assembly language, even relatively simple computations can require long and complex programs.

During the early period of computing, "programmers" ordinarily wrote programs exclusively in machine language. Today, object code is rarely written directly by computer programmers. Rather, modern programmers typically write computer programs in a "higher"-level programming language. These programs are called source programs, or source code. Although "source code" has been defined far more broadly in some of the literature in the field, and in some of the expert testimony in this case, more commonly the term "source code" refers to a computer program written in some programming language—such as FORTRAN (*FOR*Mula *TRAN*slation), COBOL (*CO*mmon *B*usiness *O*riented *L*anguage), Pascal, BASIC, or C—that uses complex symbolic

names, along with complex rules of syntax. In a typical higher-level programming language, for example, the above-described computation—that is,  $(A) + (B/C)$ —might be represented as follows:

```
A + B/C.
```

Unlike machine language, which is unique to each kind of CPU and which is executed directly by the computer, source code programming languages are universal to almost all computers. As a consequence, source code is executed indirectly. Thus, a program written in source code must be translated into the appropriate object code for execution in one type of computer, and into a different object code for execution in another type of computer. The translation can be effectuated by an "interpreter" program or by a "compiler" program. An "interpreter" program is a simultaneous translator that works in conjunction with the application program every time the application program is run, carrying out the instructions of the program one step at a time. In contrast, a "compiler" program translates the program once and for all into machine language, after which the translated program can be executed directly by the CPU without the need for any further resort to the compiler. A distinctive "interpreter" or "compiler" program is available for each type of source code programming language and each type of CPU.

A partly literal and partly pictorial manifestation of a computer program, still farther removed from direct use with the computer, is the flowchart. A flowchart is a graphic representation of a computer program that is written in symbols, rather than in bits or symbolic names, and with a syntax that is graphic rather than grammatical. See, e.g., Breyer, *The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, and Computer Programs*, 84 Harv.L.Rev. 281, 341 n. 235 (1970) (providing simple flowchart). A flowchart can be thought of as a kind of symbolic outline or schematic representation of a computer program's logic, which is written by a programmer once he or she has a conceptualization of the goals of the program. Creating a flowchart (at least, an

early draft) is thus, typically an early phase in the development of a software system, which is followed by the translation of the flowchart into source code. See Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan.L.Rev. 1045, 1051 (1989) (identifying five stages in the development of most application programs: (1) defining the desired task; (2) flowcharting; (3) encoding; (4) debugging; and (5) preparing documentation). On computers and computer programs in general, see Lewis Affidavit, ¶¶ 18-42 (Docket No. 274); Galler Declaration, ¶¶ 8-20 (Docket No. 257); A. Clapes, *Software, Copyright & Competition: The "Look and Feel" of the Law* 47-64 (1989); R. Bradbeer, P. DeBono & P. Laurie, *The Beginner's Guide to Computers* (1982); *McGraw-Hill Dictionary of Scientific and Technical Terms* (3d ed. 1984).

The parties agree, as a general proposition, that literal manifestations of a computer program—including both source code and object code—if original, are copyrightable. *Stern Electronics, Inc. v. Kaufman*, 669 F.2d 852, 855 n. 3 (2d Cir. 1982) ("written computer programs are copyrightable as literary works"); *Williams Electronics Inc. v. Artic International, Inc.*, 685 F.2d 870, 876-77 (3d Cir.1982) (object code copyrightable); *Hubco Data Products, Corp. v. Management Assistance Inc.*, 219 U.S.P.Q. 450, 454 (D.Id.1983) (same); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1243 (3d Cir.1983) (source and object code copyrightable), *cert. dismissed*, 464 U.S. 1033, 104 S.Ct. 690, 79 L.Ed.2d 158 (1984); *GCA Corp. v. Chance*, 217 U.S.P.Q. 718, 720 (N.D.Cal.1982) (same); *Midway Manufacturing Co. v. Strohon*, 564 F.Supp. 741, 750 (N.D.Ill.1983) (same); *Digital Communications Associates, Inc. v. Softklone Distributing Corp.*, 659 F.Supp. 449, 454 (N.D.Ga.1987) (same). Also, it appears that flowcharts, if sufficiently detailed and original, are entitled to copyright protection:

Flowcharts . . . are works of authorship in which copyright subsists, provided they are the product of sufficient intellectual labor to surpass the "insufficient intellectual labor hurdle". . . .

National Commission on New Technological Uses of Copyrighted Works, *Final Report and Recommendations* 43 (1978) hereinafter "*Final Report*"), reprinted in 5 *Copyright, Congress and Technology: The Public Record* (N. Henry, ed.1980), cited with approval in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1241 (3d Cir.1986), *cert. denied*, 479 U.S. 1031, 107 S.Ct. 877, 93 L.Ed.2d 831 (1987). See also *Synercom Technology, Inc. v. University Computing Co.*, 462 F.Supp. 1003, 1013 n. 5 (N.D.Tex. 1978) (although taking a very narrow view of the scope of copyrightability of nonliteral elements of computer programs, court nevertheless noted in *obiter dictum* that "it would probably be a violation to take a detailed description of a particular problem solution, such as a flowchart . . . and program such a description in computer language") (Higginbotham, J.); *Data Cash Systems, Inc. v. JS & A Group, Inc.*, 480 F.Supp. 1063, 1067 n. 4 (N.D.Ill.1979) (holding that copyright protection extends "to computer programs in their flow chart, source and assembly phases but not in their object phase," court concluded that computer program written in object code was not copyrightable), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir.1980) (program at issue not copyrightable because publication without notice forfeited any copyright protection); *Williams v. Arndt*, 626 F.Supp. 571, 578 (D.Mass.1985) (violation to take detailed prose description and program such description in source code).

Defendants vigorously dispute, however, the copyrightability of any nonliteral elements of computer programs. That is, defendants assert that only literal manifestations of computer programs are copyrightable. Plaintiff, on the other hand, maintains that copyright protection extends to all elements of computer programs that embody original expression, whether literal or nonliteral, including any original expression embodied in a program's "user interface."

One difficulty with plaintiff's argument is the amorphous nature of "nonliteral" elements of computer programs. Unlike the written code of a program or a flowchart that can be printed on paper, nonliteral elements—including such elements as the overall organization of a program, the structure

of a program's command system, and the presentation of information on the screen—may be less tangibly represented. Whether these elements are copyrightable, and if so, how the nonliteral elements that are copyrightable may be identified, are central to deciding this case.

## II. CONSTITUTIONAL CONSTRAINTS

In considering the legal issues relevant to whether nonliteral elements of Lotus 1-2-3 are copyrightable, and if so, to what extent, one may appropriately begin with a provision of the Constitution of the United States:

The Congress shall have Power . . . To promote the Progress of Science . . . by securing for limited Time to Authors . . . the exclusive Right to their . . . Writings. . . .

U.S. Const., Art. I, § 8, cl. 8. The copyright law, codified in Title 17 of the United States Code, rests upon this explicit grant of legislative authority.

Under this constitutional mandate, Congress has broad though not unlimited authority to grant copyright monopolies as needed to promote progress. If Congress were to determine, for example, that copyright protection is unnecessary to "promote the Progress of" computer programming—because, for example, in Congress' view the financial incentives alone of developing new computer programs (without the added benefit of copyright) are enough to encourage innovation, or because incremental innovation might be stifled by expansive copyright protection—then Congress could, without offending the Constitution, provide no copyright protection for computer programs. At the other extreme, were Congress to find that strong copyright protection is necessary to promote the progress of computer programming, Congress could provide for expansive copyright protection for all aspects of computer programs, again without having strayed beyond the bounds of the constitutionally permissible.

Because the constitutional grant of power authorizes Congress to take either path—or to chart some middle course—

this case does not raise constitutional issues. Rather, the issues at stake here are issues of statutory meaning. The central question is not whether Congress could render nonliteral elements such as those of 1-2-3 copyrightable, but whether it has done so. *Banks v. Manchester, Ohio*, 128 U.S. 244, 252, 9 S.Ct. 36, 39, 32 L.Ed. 425 (1888) ("No authority exists for obtaining a copyright beyond the extent to which Congress has authorized it. A copyright cannot be sustained as a right existing at common law; but, as it exists in the United States, it depends wholly on the legislation of Congress.").

## III. CONGRESSIONAL MANDATES AND JUDICIAL INTERPRETATION

### A. Sources of Guidance

#### 1. Precedent on Determining Statutory Meaning

In *Kelly v. Robinson*, 479 U.S. 36, 107 S.Ct. 353, 93 L.Ed.2d 216 (1986) (Powell, J., joined by Rehnquist, C.J., and Brennan, White, Blackmun, O'Connor, and Scalia, JJ.), the Supreme Court explained the various sources of guidance to which a court should look in determining the manifested meaning of a statute and the manifestations of congressional intent:

[T]he "starting point in every case involving construction of a statute is the language itself." *Blue Chip Stamps v. Manor Drug Stores*, 421 U.S. 723, 756 [95 S.Ct. 1917, 1935, 44 L.Ed.2d 539] (1975) (Powell, J., concurring). But the text is only the starting point. As Justice O'Connor explained last Term, " ' "In expounding a statute, we must not be guided by a single sentence or member of a sentence, but look to the provisions of the whole law, and to its object and policy." ' " *Offshore Logistics, Inc. v. Tallentire*, 477 U.S. 207, 221 [106 S.Ct. 2485, 2493, 91 L.Ed.2d 174] (1986) (quoting *Mastro Plastics Corp. v. NLRB*, 350 U.S. 270, 285 [76 S.Ct. 349, 359, 100 L.Ed. 309] (1956) (in turn quoting *United States v. Heirs of Boisdoré*, 8 How. [49 U.S.] 113, 122 [12 L.Ed. 1009] (1850))).

*Id.* at 43, 107 S.Ct. at 358. Accordingly, to determine whether Congress has extended copyright protection to nonliteral elements of computer programs, and if so, to what extent, a court must examine, first, the relevant language of the copyright statutes, second, "the provisions of the whole law," and third, "its object and policy." *Id.*

Examination of these sources exposes mandates inconsistent with the principal argument advanced by defendants—that copyright protection extends only to literal manifestations of computer programs and not to any nonliteral elements. Close examination also discloses that Congress has not explicitly addressed some of the questions that must be decided in this case. In these circumstances, it is appropriate to consider legislative history as well as statutory text.

## 2. The Statutory Language and the History of Amendments

### a. Pre-1976 Legislation

The First Congress extended copyright protection to "any map, chart, book or books already printed." Act of May 31, 1790, ch. 15, § 1, 1 Stat. 124, 124 (repealed 1831). Congress quickly expanded this limited scope of copyright protection, adding designs, prints, etchings and engravings in 1802, "musical composition" in 1831, "dramatic composition" in 1856, "photographs and the negatives thereof" in 1865, and "statuary" and "models or designs intended to be perfected as works of the fine arts" in 1870. Act of Apr. 29, 1802, ch. 36, § 2, 2 Stat. 171, 171, *repealed by* Act of Feb. 3, 1831, ch. 16, §§ 1, 14, 4 Stat. 436, 436, 439, *amended by* Act of Aug. 18, 1856, ch. 169, 11 Stat. 138, 139, *amended by* Act of Mar. 3, 1865, ch. 126, §§ 1, 2, 13 Stat. 540, 540, *repealed by* Act of July 8, 1870, ch. 180, § 86, 16 Stat. 198, 212 (repealed 1909).

In 1909, Congress abandoned the effort to list exhaustively all works in which copyright may subsist, instead adopting a more generalized approach to copyrightability:

The works for which copyright may be secured under this title shall include *all the writings of an author.*

Act of Mar. 4, 1909, ch. 320, § 4, 35 Stat. 1075, 1076 (emphasis added) (previously codified at 17 U.S.C. § 4, *reprinted in* 17 U.S.C.A. App. § 4 (West Supp.1990); recodified 1947; repealed 1976). To clarify the meaning of "all the writings of an author," Congress also provided a non-exclusive list of examples. *Id.* at § 5 (listing the various kinds of works previously entitled to explicit copyright protection). Even this generalized statement of the scope of copyrightability along with the explanatory list proved inadequate. In 1912, Congress added "motion pictures" as a further example of "all the writings of an author." Act of Aug. 24, 1912, ch. 356, § 5(1)–(m), 37 Stat. 488, 488 (previously codified at 17 U.S.C. § 5(1)–(m), *reprinted in* 17 U.S.C.A. App. § 5(1)–(m); recodified 1947; repealed 1976), and in 1972, Congress added "sound recordings" to the list, Act of Oct. 15, 1971, Pub.L. 92-140, § 1(b), 85 Stat. 391, 391 (previously codified at 17 U.S.C. § 5(n), *reprinted in* 17 U.S.C.A. App. § 5(n); repealed 1976).

In 1955, Congress began to consider another major revision of the copyright law. After twenty years of hearings, study, debate, and redrafting, that revision was signed into law in 1976. Act of Oct. 19, 1976, Pub.L. 94-553, 90 Stat. 2541 (codified at 17 U.S.C. §§ 101 *et seq.*). That law—the Copyright Act of 1976—and the 1980 amendments to the Act provide the relevant statutory mandates for this case.

### b. The Copyright Act of 1976

Like the Copyright Act of 1909, the Copyright Act of 1976 eschews the prescription of an exclusive list of the kinds of works that are copyrightable:

Copyright protection subsists, in accordance with this title, in *original works of authorship* fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.

17 U.S.C. § 102(a) (1988) (emphasis added).

It is axiomatic that the designation "original" is not intended to be limited to works that are novel or unique. Rather, the word "original," which was "purposely left undefined" by Congress, refers to works that have been "independently created by an author," regardless of their literary or aesthetic merit, or ingenuity, or qualitative value. H.R.Rep. No. 1476, 94th Cong., 2d Sess. 51, *reprinted at* 1976 U.S. Code Cong. & Admin. News 5659, 5664 (hereinafter "House Report"); *Hutchinson Telephone Co. v. Fronteer Directory Co.*, 770 F.2d 128, 131 (8th Cir.1985); *Puddu v. Buonamici Statuary, Inc.*, 450 F.2d 401, 402 (2d Cir.1971) ("originality has been considered to mean 'only that the work owes its origin to the author' ") (emphasis added; *quoting* Nimmer, *Copyright* § 10 at 32 (1971 ed.)). *See also* *Bleistein v. Donaldson Lithographing Co.*, 188 U.S. 239, 251, 23 S.Ct. 298, 300, 47 L.Ed. 460 (1903) (Holmes, J.) ("It would be a dangerous undertaking for persons trained only in the law to constitute themselves final judges of the worth of" a work.).

Also, the designation "works of authorship" is not meant to be limited to traditional works of authorship such as novels or plays. Rather, Congress used this phrase to extend copyright to new methods of expression as they evolve:

The history of copyright law has been one of gradual expansion in the types of works accorded protection, and the subject matter affected has fallen into two general categories. In the first, scientific discoveries and technological developments have made possible new forms of creative expression that never existed before. In some of these cases the new expressive forms—electronic music, filmstrips, and *computer programs*, for example—could be regarded as an extension of copyrightable subject matter Congress had already intended to protect, and were thus considered copyrightable from the outset without need of new Legislation. In other cases, such as photographs, sound recordings, and motion pictures, statutory enactment was deemed necessary to give them full recognition as copyrightable works.

Authors are continually finding new ways of expressing themselves, but it is impossible to foresee the forms that these new expressive methods will take. The bill does not intend either to freeze the scope of copyrightable subject matter at the present stage of communications technology or to allow unlimited expansion into areas outside the present congressional intent.

House Report at 51, *reprinted at* 5664 (emphasis added).

To help illumine the meaning of "works of authorship," Congress, as it had done in the 1909 Act, again provided a statutory list of examples of those kinds of works that, if original, merit copyright protection:

Works of authorship include the following categories: (1) literary works; (2) musical works, including any accompanying words; (3) dramatic works, including any accompanying music; (4) pantomimes and choreographic works; (5) pictorial, graphic, and sculptural works; (6) motion pictures and other audiovisual works; and (7) sound recordings.

17 U.S.C. § 102(a) (1988). This listing was intended to be " 'illustrative and not limitative,' and . . . the seven categories do not necessarily exhaust the scope of 'original works of authorship' that the bill is intended to protect." House Report at 53, *reprinted at* 5666. Consequently, in addition to these explicitly-listed items, courts have extended copyright protection to such works as artistic features of masquerade costumes, *National Theme Productions, Inc. v. Jerry B. Beck, Inc.*, 696 F.Supp. 1348, 1354 (S.D.Cal.1988); the arrangement of public-domain legal decisions in reporters, *West Publishing Co. v. Mead Data Central, Inc.*, 799 F.2d 1219 (8th Cir.1986), *cert. denied*, 479 U.S. 1070, 107 S.Ct. 962, 93 L.Ed.2d 1010 (1987); telephone books, *Southern Bell Telephone & Telegraph Co. v. Associated Telephone Directory Publishers*, 756 F.2d 801 (11th Cir.1985); televised news reports, *Pacific and Southern Co. v. Duncan*, 744 F.2d 1490, 1494, *reh'g denied*, 749 F.2d 733 (11th Cir.1984), *cert. denied*, 471 U.S. 1004, 105 S.Ct. 1867, 85 L.Ed.2d 161

(1985); blank answer sheets for use with student achievement and intelligence tests that are designed to be corrected by optical scanning machines, *Harcourt, Brace & World, Inc. v. Graphic Controls Corp.*, 329 F.Supp. 517 (S.D.N.Y.1971) (applying the 1909 Act); maps, *C.S. Hammond & Co. v. International College Globe, Inc.*, 210 F.Supp. 206, 216 (S.D.N.Y. 1962) (applying the 1909 Act); and a code book for cable correspondence consisting of 6,325 coined, otherwise meaningless, words of five letters each, *Reiss v. National Quotation Bureau, Inc.*, 276 F. 717 (S.D.N.Y.1921) (L. Hand, J.) (applying the 1909 Act).

Although Congress did not include "computer programs" in this list of examples of "works of authorship," computer programs fall squarely within the statutory definition of literary works:

"Literary works" are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.

17 U.S.C. § 101 (1988). See also House Report at 54, reprinted at 5667 ("[t]he term 'literary works' . . . includes . . . computer programs") *id.* at 51, reprinted at 5664 ("computer programs . . . were . . . considered copyrightable from the outset"); *id.* at 116, reprinted at 5731 (1976 Act governs "copyright-ability [sic] of computer programs").

Like all other works of authorship, however, computer programs, even if certain elements of them are copyrightable, are not entitled to an unlimited scope of copyright protection. Most relevant to this case is the following limitation:

In no case does copyright protection for an original work of authorship extend to an *idea, procedure, process, system, method of operation, concept, principle, or discovery*, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

17 U.S.C. § 102(b) (1988) (emphasis added). Noting that this section applies to computer programs, the House Report declares: "Section 102(b) is intended, among other things, to make clear that the *expression* adopted by the programmer is the copyrightable element in a computer program, and that the actual *processes or methods* embodied in the program are not within the scope of the copyright law." *House Report* at 57, reprinted at 5670 (emphasis added); see also *id.* at 54, reprinted at 5667 (computer programs are copyrightable only "to the extent that they incorporate authorship in the programmer's *expression* of original ideas, as distinguished from the *ideas* themselves") (emphasis added).

#### c. CONTU and the 1980 Amendments

Most contemporaneous observers expected that the bill that eventually became the Copyright Act of 1976 would apply to computer programs. Toward the end of the twenty-year process of reconsidering the copyright law, however, Congress recognized that certain problems raised by computer and other new technologies were not adequately addressed in the pending bill. See House Report at 116, reprinted at 5731. Accordingly, in 1974, Congress created the National Commission on New Technological Uses of Copyrighted Works ("CONTU"). Congress gave the Commission the following mandate:

(b) The purpose of the Commission is to study and compile data on:

(1) the reproduction and use of copyrighted works of authorship—

(A) in conjunction with automatic systems capable of storing, processing, retrieving, and transferring information . . . .

(c) The Commission shall make recommendations as to such changes in copyright law or procedures that may be necessary to assure for such purposes access to copyrighted works, and to provide recognition of the rights of copyright owners.

Act of Dec. 31, 1974, Pub.L. 93-573, § 201(b)-(c), 88 Stat. 1873, 1873-74 (1974).

It took seven months after enactment of the bill to constitute the Commission and appoint the Commissioners. Because of this delay, CONTU did not begin its deliberations until October 1975, and did not release its report and recommendations until July 1978, almost two years after the passage of the 1976 Act.

CONTU observed a need for copyright protection of creative expression embodied in computer programs:

The cost of developing computer programs is far greater than the cost of their duplication. Consequently, computer programs . . . are likely to be disseminated only if . . . [t]he creator can spread its costs over multiple copies of the work with some form of protection against unauthorized duplication of the work. . . . The Commission is, therefore satisfied that some form of protection is necessary to encourage the creation and broad distribution of computer programs in a competitive market, . . . [and] that the continued availability of copyright protection for computer programs is desirable.

CONTU, *Final Report* at 20-21. Concluding, however, that the Act of 1976 already provided adequate protection, CONTU did not propose any statutory changes with respect to *copyrightability* of computer programs. On the other hand, CONTU did propose two amendments with respect to *permissible copying* of computer programs. These proposed amendments were subsequently adopted by Congress with only minor modifications and with little additional legislative history.

First, tracking verbatim CONTU's recommendation, Congress amended section 101 to include the following definition:

A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

Act of Dec. 12, 1980, Pub.L. 96-517, 94 Stat. 3015, 3028 (codified at 17 U.S.C. § 101). Second, Congress also followed CONTU's recommendation by amending section 117 to allow the owner of a computer program to make additional copies or adaptations of the program:

Notwithstanding the provisions of section 106 [which grants the copyright owner the exclusive rights to reproduce the copyrighted work], it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:

(1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or

(2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.

*Id.* (codified at 17 U.S.C. § 117).

Finally, although the Commission did not address explicitly the central issue of the present case—whether copyright protection extends to the nonliteral elements of computer programs at issue here—CONTU did re-emphasize, for purposes of copyrightability of computer programs, the fundamental distinction between copyrightable expression on the one hand, and noncopyrightable methods, processes and ideas on the other. *Id.* at 37-46. Indeed, although his personal views are entitled to very little if any weight in the context of the

court's determination of the statutory mandates, *Whelan*, 797 F.2d at 1241, n. 37, it is interesting to note that Melville Nimmer, Vice-Chairperson of CONTU, testified about the Commission's intent. in *Evergreen Consulting v. NCR Comten, Inc.*, No. 82-5946-KN (C.D.Cal. filed 1982). According to Nimmer, CONTU understood that the proposed repeal of former section 117 would extend copyright protection to non-literal elements of computer programs:

CONTU had no views, and made no recommendations which would negate the availability of copyright protection for the detailed design, structure and flow of a [computer] program under the copyright principles that make copyright protection available, in appropriate circumstances, for the structure and flow of a novel, a play or a motion picture.

Nimmer Decl. in *Evergreen Consulting*, ¶ 28, reprinted in Note, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 Mich.L.Rev. 866, 889 (1990). But cf. conflicting views of Commissioner Arthur Miller and Executive Director Arthur Levine, discussed in *id.* at 888-90.

### 3. Relevant Aspects of the Whole Law of Copyright

#### a. "Nonliteral" Expression

With respect to such things as musical, dramatic, and motion picture works, and works of "literature" (as contrasted with "literary" works in the broader statutory sense, see Samuelson, *Reflections on the State of American Software Copyright Law and the Perils of Teaching It*, 13 Colum.-VLA J.L. & Arts 61, 65 n. 15 (1988)), it is crystal clear that, to the extent original, the literal manifestations of such works are protected by copyright. Thus, during a period of copyright protection, one cannot copy an author's book, score, or script without authorization in law or in fact. It is also well settled that a copyright in a musical, dramatic, or motion picture work, or a work of literature, may be infringed even if the infringer has not copied the literal aspects of the work. That

is, even if an infringer does not copy the words or dialogue of a book or play, or the score of a musical work, infringement may be found if there is copying of the work's expression of setting, characters, or plot with a resulting substantial similarity. *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir.1930) (L. Hand, J.) ("the right cannot be limited literally to the text, else a plagiarism would escape by immaterial variations"). *cert. denied*, 282 U.S. 902, 51 S.Ct. 216, 75 L.Ed. 795 (1931); *Sheldon v. Metro-Goldwyn Pictures Corp.*, 81 F.2d 49, 55 (2d Cir.) (L. Hand, J.) ("a play may be pirated without using the dialogue"), *cert. denied*, 298 U.S. 669, 56 S.Ct. 835, 80 L.Ed. 1392 (1936); *Detective Comics, Inc., v. Bruns Publications, Inc.*, 111 F.2d 432, 433 (2d Cir.1940) (defendant's comic book "Wonderman" adjudged to infringe plaintiff's copyright in the comic book series "Superman" where both comic books' central characters have miraculous strength and speed; conceal their strength, along with their skin-tight acrobatic costumes, beneath ordinary clothing; are termed champions of the oppressed; crush guns; stop bullets; and leap over or from buildings); *Bradbury v. Columbia Broadcasting System, Inc.*, 287 F.2d 478, 482-84 (9th Cir.) (twenty-two nonliteral similarities between plaintiff's *Fahrenheit 451* and *The Fireman* and defendants' television production), *cert. dismissed*, 368 U.S. 801, 82 S.Ct. 19, 7 L.Ed.2d 15 (1961); *Roth Greeting Cards v. United Card Co.*, 429 F.2d 1106 (9th Cir.1970) (substantial similarity between nonliteral expressive elements embodied in defendant's cards and plaintiff's cards); *Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp.*, 562 F.2d 1157 (9th Cir.1977) (defendant's television commercial substantially similar in locale, characters, and plot to plaintiff's children's television series); *Twentieth Century-Fox Film Corp. v. MCA, Inc.*, 715 F.2d 1327 (9th Cir.1983) (thirteen alleged similarities between plaintiff's motion picture, *Star Wars*, and defendant's motion picture and derivative television series, *Battlestar: Gallactica*—including totality of setting, characters and their relationships, and elements of plot—were sufficient to create genuine dispute of material fact to defeat

defendant's motion for summary judgment); *Horgan v. Macmillan, Inc.*, 789 F.2d 157, 162 (2d Cir.1986) (copyright in choreography for *The Nutcracker* ballet may be infringed by a book of photographs of the ballet if the series of photographs is substantially similar to the ballet). *See also Stewart v. Abend*, \_\_\_ U.S. \_\_\_, 110 S.Ct. 1750, 1759, 109 L.Ed.2d 184 (1990) (noting that a motion picture may infringe a book by using "the story's unique setting, characters, plot, and sequence of events"). This type of copying of nonliteral expression, if sufficiently extensive, has never been upheld as permissible copying; rather, it has always been viewed as copying of elements of an expression of creative originality.

b. "Useful Articles"

A "useful article" is defined by the Copyright Act as

an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information. An article that is normally a part of a useful article is considered a "useful article."

17 U.S.C. § 101 (1988). Such articles—or more accurately, the utilitarian aspects of such articles—are not works of authorship in which copyright can subsist. House Report at 55, *reprinted at* 5668. However:

[T]he design of a useful article . . . shall be considered a pictorial, graphic, or sculptural work [which is copyrightable] . . . if, and only to the extent that, such design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article.

17 U.S.C. § 101 (1988) (defining "pictorial, graphic, and sculptural works"). Put more broadly, the point is that those elements of a useful article that can exist independently of the utilitarian aspects of the article are potentially copyrightable because those elements are elements of *expression* that can be distinguished from the utilitarian functions of the article. *See*

*Whelan*, 797 F.2d at 1236 ("the purpose or function of a utilitarian work [is] the work's idea, and everything that is not necessary to that purpose or function [is] part of the expression of the idea"); *Williams Electronics, Inc. v. Bally Manufacturing Corp.*, 568 F.Supp. 1274, 1279-80 (N.D. Ill.1983) (Congress intended to exclude from copyright protection functional elements of work, leaving only those aspects of work separable from utilitarian aspects copyrightable).

#### 4. The Objects and Policies of Copyright Law

The court's final task in divining the statutory mandates is to look to the "object and policy" of the copyright law. *Kelly*, 479 U.S. at 43, 107 S.Ct. at 358. This inquiry has received heightened attention in this case (*see* Part VII, *infra*) because defendants contend that extending copyright protection to nonliteral elements of computer programs is contrary to the objects and policies of copyright law as expressed in the copyright statute and in precedents. Although, of course, disputing defendants' conclusion, plaintiff does not contest the premise of this contention—that in construing the manifested meaning of the Copyright Act, the court is directed to look to the "object and policy" of the copyright law.

Copyright monopolies are not granted for the purpose of rewarding authors. Rather, Congress has granted copyright monopolies to serve the public welfare by encouraging authors (broadly defined) to generate new ideas and disclose them to the public, being free to do so in any uniquely expressed way they may choose. *Harper & Row, Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 546, 105 S.Ct. 2218, 2223, 85 L.Ed.2d 588 (1985). As the Supreme Court has concluded, "encouragement of individual effort by personal gain is the best way to advance public welfare through talents of authors and inventors in 'Science and useful Arts.'" *Mazer v. Stein*, 347 U.S. 201, 219, 74 S.Ct. 460, 471, 98 L.Ed. 630 *reh'g denied*, 347 U.S. 949, 74 S.Ct. 637, 98 L.Ed. 1096 (1954). *See also Sony Corp. v. Universal City Studios, Inc.*, 464 U.S. 417, 432, 104 S.Ct. 774, 783, 78 L.Ed.2d 574 (1984) ("The immediate effect of our copyright

law is to secure a fair return for an "author's" creative labor. But the ultimate aim is, by this incentive, to stimulate artistic creativity for the general public good.' ") (quoting *Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151, 156, 95 S.Ct. 2040, 2043, 45 L.Ed.2d 84 (1975)).

In construing the relevant statutory mandates, the court must be faithful to the statutory language and mindful of both the ultimate goal of copyright law—the advancement of public welfare—and Congress' chosen method of achieving this goal—private reward to the individual author. Courts should not draw the line between copyrightable and non-copyrightable elements of computer programs in such a way as to harm the public welfare, nor should courts ignore the accommodation struck by Congress in choosing to advance the public welfare by rewarding authors. These mandates leave courts with a delicate task:

Drawing the line too liberally in favor of copyright protection would bestow strong monopolies over specific applications upon the first to write programs performing those applications and would thereby inhibit other creators from developing improved products. Drawing the line too conservatively would allow programmer's efforts to be copied easily, thus discouraging the creation of all but modest incremental advances.

Menell, *Scope of Copyright Protection for Programs*, 41 Stan.L.Rev. at 1047-48. See also Note, *Scope of Copyright Protection of Computer Program Structure*, 88 Mich.L.Rev. at 895; Note, *Defining the Scope of Copyright Protection for Computer Software*, 38 Stan.L.Rev. 497, 498 (1986).

Rather than itself drawing the boundary line between copyrightable and non-copyrightable elements of computer programs, Congress has mandated that courts use an evaluative standard in determining this boundary line—that is, a standard that distinguishes idea from expression and requires that a court, in applying this distinction, be sensitive to the object and policy of copyright law as manifested by Congress.

When statutes establish evaluative standards for deciding cases, courts—by necessity—must locate boundaries in uncharted terrain, using the markers that Congress has placed. At some places a boundary may run straight as a surveyor's sight-line between markers; at others, it may meander like a stream, moving toward resolution of clashing objects and policies the markers identify. No marker Congress has placed may be disregarded or relocated by courts. Even in those instances where text and context make clear that literal description of a marker is contrary to manifested meaning (as where "not" must be inserted or deleted to make sense of the statutory language) what the court is doing is aptly described as "locating," not "relocating," the marker Congress mandated. The fewer the markers Congress has placed, the more critical it becomes that courts assure that no marker escapes notice.

#### B. *The Idea-Expression and Useful-Expressive Distinctions*

Although the statutory mandates are ambiguous in some critical respects, one point on which they are clear (one marker of the boundary line) is this: at least some, but clearly not all, aspects of computer programs, if original, are "works of authorship" in which copyright can subsist. 17 U.S.C. § 102(a)(1); House Report at 54, reprinted at 5667; CONTU, *Final Report* at 21. How can a court determine which aspects are copyrightable?

The interplay between sections 102(a) and 102(b), illumined by the related legislative history, manifests that the statute extends copyright protection to expressive elements of computer programs, but not to the ideas, processes, and methods embodied in computer programs. House Report at 54, 57, reprinted at 5667, 5670. This dichotomy—which is often referred to as the "idea-expression distinction," and which embraces also the process-expression, method-expression, and useful-expressive distinctions, see Note, *Determining the Scope of Copyright Protection of Computer Program Structure*, 88 Mich.L.Rev. at 866-67—has long been a fundamental part of our copyright law. *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879). In that seminal case, the Court held that the

text of a book describing a special method of double-entry accounting on paper spreadsheets—the now almost universal T-accounts system—was copyrightable *expression*, but that the method itself, which embodied the *idea* of this particular kind of double-entry bookkeeping, was not. The Court thus concluded that Baker did not infringe Selden's copyright when Baker wrote his own treatise, in his own words, describing the special double-entry method of bookkeeping.

CONTU, too, concluded that the idea-expression distinction should be used to determine which aspects of computer programs are copyrightable. CONTU, *Final Report* at 37-46. The Commission recognized, though, that “[i]t is difficult, either as a matter of legal interpretation or technological determination, to draw the line between the copyrightable element of style and expression in a computer program and the process which underlies it.” *Id.* at 44.

CONTU, of course, was not an official voice of Congress, and its views are not, without more, attributable to Congress. Thus, courts must not treat the CONTU report as legislative history, in the ordinary sense, much less as an authoritative statement about manifested legislative intent. *Whelan*, 797 F.2d at 1241-42. The privately held views of Vice-Chairperson Nimmer and Commissioner Miller, and especially the privately held views of Executive Director Levine, *see* declarations in *Evergreen Consulting*, without some manifestation of congressional endorsement, are even less relevant to the determination of manifested congressional intent. *Id.* at 1241 n. 37.

Congress, however, did not ignore CONTU. Indeed, as already noted, Congress adopted practically verbatim the Commission's proposed statutory changes with respect to computer programs. Thus, the expressed views of the Commission, to the extent not repudiated by Congress, may help to explain the context in which Congress acted, which in turn may support inferences about the meaning of any otherwise ambiguous passages in what Congress declared.

I conclude that, with the Copyright Act of 1976 and the 1980 amendments to that Act, Congress manifested an inten-

tion to use the idea-expression distinction as part of the test of copyrightability for computer programs. That is, rather than adopting some other test of copyrightability that made the idea-expression dichotomy irrelevant (such as defendants' proposed literal-nonliteral distinction), Congress chose to extend copyright protection to original expression embodied in computer programs, but not to any idea, method, or process described by that expression. 17 U.S.C. §§ 102(a), 102(b); House Report at 54, 57, *reprinted at* 5667, 5670.

That conclusion is consistent with the treatment of other kinds of intellectual works—specifically, with the treatment of nonliteral elements of expression in musical, dramatic, and motion picture works, and works of literature. It is also consistent with the treatment of useful articles. That is, as explained more fully in Part IV(A), *infra*, I conclude that the user interface and some other nonliteral aspects of computer programs are *not merely* articles “having an intrinsic utilitarian function.” 17 U.S.C. § 101 (defining “useful article”). When computer programs include elements—both literal and nonliteral—“that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article,” *id.* (defining “pictorial, graphic, and sculptural works”), they are potentially copyrightable.

Finally, this conclusion is also consistent with the objects and policies of copyright—to encourage the creation and dissemination of new *ideas* by protecting, for limited times, the specific way that an author has *expressed* those ideas.

#### IV. THE LEGAL TEST FOR COPYRIGHTABILITY APPLICABLE TO THIS CASE

##### A. *Functionality, Useful Articles, and the Useful-Expressive Distinction*

Defendants suggest that the user interface of Lotus 1-2-3 is a useful, “function[al]” object like the functional layout of gears in an “H” pattern on a standard transmission, the functional assignment of letters to keys on a standard QWERTY keyboard, and the functional configuration of controls on a

musical instrument (e.g., keys of a piano). Lewis Affdvt. ¶¶ 52-54. These "functional" "useful articles," defendants contend, are not entitled to copyright protection.

A similar analogy was made in *Synercom* where the court concluded that a sequence of data inputs for a statistical analysis program was like the "figure-H" pattern of a standard transmission. 462 F.Supp. at 1013. *Synercom*, though, was published less than a month after the publication of the CONTU report (which it never cites) and well before the 1980 amendments. Since then, congressional and judicial development of the law of copyrightability of computer programs has advanced considerably, and *Synercom*'s central proposition—that the expression of nonliteral sequence and order is inseparable from the idea and accordingly is not copyrightable—has been explicitly rejected by several courts. E.g., *Whelan*, 797 F.2d at 1240, 1248 ("copyright protection of computer programs may extend beyond the programs' literal code to their structure, sequence, and organization"); *Broderbund Software, Inc. v. Unison World, Inc.*, 648 F.Supp. 1127, 1133 (N.D.Cal.1986) ("copyright protection is not limited to the literal aspects of a computer program, but rather . . . it extends to the overall structure of a program, including its audiovisual displays"). Moreover, even those courts that have not gone as far as *Whelan* and *Broderbund* have still gone much farther in protecting computer programs than *Synercom*. E.g., *SAS Institute, Inc. v. S & H Computer Systems, Inc.*, 605 F.Supp. 816, 830 (M.D.Tenn.1985) ("copying of the organization and structural details" can form basis for infringement); *Manufacturers Technologies, Inc. v. CAMS, Inc.*, 706 F.Supp. 984, 993 (D.Conn.1989) ("screen displays or user interface" copyrightable); *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d 1173, 1175 (9th Cir.1989) (nonliteral aspects such as "structure, sequence and/or organization of the program, the user interface, and the function, or purpose, of the program", are copyrightable to the extent that they embody expression rather than idea); *Telemarketing Resources v. Symantec Corp.*, 12 U.S.P.Q.2d 1991, 1993, 1989 WL 200350 (N.D. Cal.1989) (holding that "[c]opyright pro-

tection applies to the user interface, or overall structure and organization of a computer program, including its audiovisual displays, or screen 'look and feel,' " but finding no infringement in this case); *Q-Co. Industries v. Hoffman*, 625 F.Supp. 608, 615-16 (S.D.N.Y.1985) (similarity of "structure and arrangement" can form basis of infringement suit, but here, structural similarities were dictated by functional considerations and hence were non-copyrightable ideas rather than copyrightable expression); *Pearl Systems, Inc. v. Competition Electronics, Inc.*, 8 U.S.P.Q.2d 1520, 1524, 1988 WL 146047 (S.D.Fla.1988) ("Copyright protection of computer software is not limited to the text of the source or object code"). But see *Softklone*, 659 F.Supp. at 455, 465 (rejecting *Broderbund*'s conclusion that audiovisual screen displays are copyrightable, although holding that separate copyright on status screen display was infringed where "total concept and feel" was copied); *Plains Cotton Cooperative Association v. Goodpasture Computer Service, Inc.*, 807 F.2d 1256, 1262 (rejecting *Whelan*'s protection for structure, sequence, and organization, court instead held that sequence and organization, where dictated by market forces, is non-copyrightable idea rather than copyrightable expression), *reh'g denied*, 813 F.2d 407 (5th Cir.), *cert. denied*, 484 U.S. 821, 108 S.Ct. 80, 98 L.Ed.2d 42 (1987). In any event, *Synercom*'s input formats are quite different from, and distinguishable from, the nonliteral aspects of 1-2-3 at issue in this case.

Defendant's proposed analogy is also similar to the analogy drawn by Commissioner Hersey between a computer program and an object that is designated to do work—for example, the cam of a drill. CONTU, *Final Report* at 58-60 (Hersey, C., dissenting). His view, however, was in dissent, and not a view advanced by CONTU. Because Congress adopted CONTU's recommendations practically verbatim, it is reasonable to infer that Congress did not adopt Commissioner Hersey's view.

Moreover, I conclude that defendants' contentions, to the extent they are similar to *Synercom*'s central proposition and to Commissioner Hersey's views in dissent, are inconsistent

with the legislative history and statutory mandates explained above. If, in a context such as that of *Synercom* or of this case, an idea and its expression were taken to be inseparable and the expression therefore not copyrightable, copyright law never would, as a practical matter, provide computer programs with protection as substantial as Congress has mandated—protection designed to extend to original elements of expression however embodied. I credit the testimony of expert witnesses that the bulk of the creative work is in the conceptualization of a computer program and its user interface, rather than in its encoding, and that creating a suitable user interface is a more difficult intellectual task, requiring greater creativity, originality, and insight, than converting the user interface design into instructions to the machine. *See* Galler Decl. ¶¶ 37, 39; Emery Decl. ¶¶ 22, 25-28 (Docket No. 265); Reed Decl. ¶¶ 26-27 (Docket No. 290). Defendants' contentions would attribute to the statute a purpose to protect only a narrowly defined segment of the creative development of computer programs, and to preclude from protection even more significant creative elements of the process. Such a result is fundamentally inconsistent with the statutory mandates.

Also, defendants' contention would have the additional consequence that computer programmers would have little, if any, more protection for nonliteral elements of expression embodied in their original works of authorship than is already provided by trade secret law. If the intellectual effort and creativity embodied in a user interface were protectable only by trade secret law, the length of protection for computer programs would be very short—merely the time it takes to examine a program and then duplicate the nonliteral elements in a newly written computer program. This short period of protection is fundamentally inconsistent with the mandates of the copyright law.

Defendants have advanced their "useful article" (or "functionality") argument in many forms. Never, however, have they stated every premise that is a step of the reasoning implicit in the argument. One set of unstated premises is the

assumed meanings of "useful," "article," and "useful article." Indeed, different forms of their argument have depended on different assumed meanings for one or more of these three terms. Some illustrations will clarify this point.

Suppose, first, the form of the argument is this:

A "useful article" is not copyrightable; a "computer program" is an "article," and a good "computer program" is "useful"; therefore, a good "computer program" is not copyrightable.

In this illustration, "useful," "article," and "useful article" are assumed to have meanings consistent with ordinary usage (with an exception to be noted below). The proposition that emerges when the words are interpreted in this way is, of course, plainly contrary to congressional mandates. Congress must have had some purpose in defining "computer program" and declaring that an "owner of a copy of a computer program" may make additional copies, in specified circumstances, without violating copyright laws. 17 U.S.C. §§ 101, 117. The clear implication of the 1980 amendments is that in some circumstances, at least, a good computer program *is* copyrightable. Otherwise, there would have been no need for Congress to enact these two provisions about a "computer program" and additional "copies."

It follows that, for this defense argument to make any sense and also be consistent with congressional mandates, defendants must be proposing that "useful," or "article," or "useful article," or all three, be interpreted in some sense different from their ordinary meanings. In what sense, though, they never say.

Suppose, second, a different form of the argument:

A "useful article" is not copyrightable; a "screen display" is an "article," and a good "screen display" is "useful"; therefore, a good "screen display" is not copyrightable.

Here, the implicit meaning of "article" is being stretched beyond ordinary usage to apply to what one would ordinarily think of as just one part of an article, rather than itself an arti-

cle. Also, the implicit meaning of "article" stretches beyond what one would ordinarily think of as something that, as part of a larger "article," is itself an "article." Thus, the implicit meaning of "article" is broader still than the statutory mandate that "[a]n *article* that is normally a part of a useful article is considered a 'useful article.'" 17 U.S.C. § 101 (emphasis added). Are these expansions of the meaning of "article" consistent with the text and with the object and policy manifested in the statute? We may be better prepared to answer after considering one more illustrative form of the argument.

Suppose, third, this form of the argument:

A "useful article" is not copyrightable; a "user interface" is an "article," and a good "user interface" is "useful"; therefore, a good "user interface" is not copyrightable.

Here, the meaning of "article" is plainly stretched beyond ordinary usage, and especially when we understand "user interface" in a sense broad enough to include nonliteral elements such as the command structure. Moreover, unease that we are being led into a departure from statutory text and manifested object and policy grows deeper as we reflect still more on this illustrative form of the argument.

One problem is that the argument depends on changing the assumed meaning of one or all of "useful," "article," and "useful article," to suit the needs of the copier who is advancing the argument as a defense against a claim of copyright infringement.

An even deeper fallacy of the argument, however, is that it assumes a meaning of "useful article" in step one ("a 'useful article' is not copyrightable") that goes far beyond ordinary meanings of "useful" and "article" combined. That is, not merely does it assume a *descriptive* meaning of something that is an "article" and is also "useful," but in addition it assumes that the definition of "useful article" includes, as something built into it, a rule of law: Everything that is a "useful article" in a descriptive sense is also, by rule of law, not copyrightable, and under this rule of law, nothing about a

"useful article"—no element, no aspect, no part of it—can ever be copyrighted. Merely to expose this assumed prescriptive meaning of "useful article," as defendants persistently use the phrase, is to demonstrate that any argument founded upon this meaning is fundamentally inconsistent with the mandates of the copyright statute.

It may be quite true, with respect to "useful articles"—indeed I believe it to be so—that their utilitarian aspects are not copyrightable, and that things that *merely* utter work, such as the cam of a drill, are not copyrightable. It is not true, however, that every aspect of a user interface that is "useful" is therefore not copyrightable. For example, Lotus 1-2-3 is surely "useful." It does not follow that when an intellectual work achieves the feat of being useful as well as expressive and original, the moment of creative triumph is also a moment of devastating financial loss—because the triumph destroys copyrightability of all expressive elements that would have been protected if only they had not contributed so much to the public interest by helping to make some article useful.

Defendants' contention misses this point by proceeding on an erroneous assumption about the role of "functionality" in copyright law. It is true that "functionality" of an article does not itself support copyrightability. Thus, it never strengthens a claim for copyright to show that the "work" for which copyright protection is claimed is useful. A congressional mandate that "proof that an intellectual work is a 'useful article' does not support the author's claim for copyright" is not, however, a mandate that "if one who copied the author's work proves that the work was 'useful' or 'functional,' the author loses *all* copyright protection." Transforming a mandate that "proof of usefulness *does not strengthen* a copyright claim" to a mandate that "proof of usefulness *destroys* a copyright claim" is, to say the least, a remarkable intellectual leap. Defendants have not advanced such a proposition explicitly. But this is in fact an implicit premise of their contention—or a consequence of it, if one takes a hindsight view of having applied their proposed rule in decisionmaking. In effect, their proposed rule would work this way: Anything that is useful is a "use-

ful article"; nothing about a "useful article" is ever copyrightable; because 1-2-3 is useful, and is an article, it is not copyrightable.

A more sensible interpretation of the statutory mandate is that the mere fact that an intellectual work is useful or functional—be it a dictionary, directory, map, book of meaningless code words, or computer program—does not mean that none of the elements of the work can be copyrightable. Also, the statute does not bar copyrightability merely because the originality of the expression becomes associated, in the marketplace, with usefulness of the work to a degree and in dimensions not previously achieved by other products on the market. *Brandir International, Inc. v. Cascade Pacific Lumber Co.*, 834 F.2d 1142, 1147 (2d Cir.1987) ("[A] copyrighted work . . . does not lose its protected status merely because it subsequently is put to functional use."); *NEC Corp. v. Intel Corp.*, 645 F.Supp. 590, 595 (N.D.Cal.1986) ("function performed by defendant's microprograms . . . does not affect their status as copyrightable subject matter"), *vacated on grounds of judge's recusal*, see 835 F.2d 1546 (9th Cir.1988). To hold otherwise would be to deny copyright protection to the most original and least obvious products of the creative mind merely because the marketplace accepts them as distinctively "functional." Such a rule would grant copyright protection for only those products that fall far short of being the best available. Rather than promoting and encouraging both the development and disclosure of the best, such a rule would offer incentives to market only the second, or third or tenth best, and hold back the best for fear that it is too good for copyrightability. Copyrightability is not a synonym for imperfection.

Accordingly, I conclude that a court, in determining whether a particular element is copyrightable, must not allow one statutory mandate—that functionality or usefulness is not itself a basis for copyrightability—to absorb and destroy another statutory mandate—that elements of expression are copyrightable. Elements of expression, even if embodied in useful articles, are copyrightable if capable of identification

and recognition independently of the functional ideas that make the article useful. This mandate may be viewed as a corollary of the central distinction of copyright law between idea and expression, which is explored further immediately below.

#### *B. The Idea-Expression Riddle: Four Additional Concepts*

It is by now plain that an idea is not copyrightable and an expression may be. It does not follow, though, that every expression of an idea is copyrightable. To begin, to get an understanding of the legally significant contrasts among an idea, noncopyrightable expressions of the idea, and a copyrightable expression, we must take account of four more concepts.

Earlier parts of this Opinion refer to two of these four—"originality" and "functionality." The expression of an idea is copyrightable only if it is original—that is, if the expression originated with the author. 17 U.S.C. § 102(a); see Part III(A)(2)(b), *supra*. Even then the expression of the idea is not copyrightable if the expression does no more than embody elements of the idea that are functional in the utilitarian sense. 17 U.S.C. § 102(b); see Part IV(A) *supra*.

The third concept is "obviousness." When a particular expression goes no farther than the obvious, it is inseparable from the idea itself. Protecting an expression of this limited kind would effectively amount to protection of the idea, a result inconsistent with the plain meaning of the statute. *E.H. Tate Co. v. Jiffy Enterprises, Inc.*, 16 F.R.D. 571, 573 (E.D.Pa.1954) (small sketch and accompanying instruction "Apply hook to wall" so obvious that it is not entitled to copyright protection).

It is only a slight extension of the idea of "obviousness"—and one supported by precedent—to reach the fourth concept: "merger." If a particular expression is one of a quite limited number of the possible ways of expressing an idea, then, under this fourth concept, the expression is not copyrightable:

When the uncopyrightable subject matter is very narrow, so that "the topic necessarily requires," if not only one

form of expression, at best only a limited number, to permit copyrighting would mean that a party or parties, by copyrighting a mere handful of forms, could exhaust all possibility of future use of the substance. In such circumstances it does not seem accurate to say that any particular form of expression comes from the subject matter. However, it is necessary to say that the subject matter would be appropriated by permitting the copyrighting of its expression. We cannot recognize copyright as a game of chess in which the public can be checkmated.

*Morissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir.1967) (citations omitted). See also *Concrete Machinery Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 606 (1st Cir.1988) ("When there is essentially only one way to express an idea, the idea and its expression are inseparable and copyright is no bar to copying that expression."); *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir.1971) (idea of a jewel-encrusted life-like bee pin inseparable from expression; thus expression not copyrightable because "protecting the 'expression' in such circumstances would confer a monopoly of the 'idea' upon the copyright owner"). Cf *Atari, Inc. v. North American Philips Consumer Electronics Corp.*, 672 F.2d 607, 616 (7th Cir.), cert. denied, 459 U.S. 880, 103 S.Ct. 176, 74 L.Ed.2d 145 (1982) (*scènes à faire* of literary works—"stock literary devices" such as "incidents, characters or settings which are as a practical matter indispensable, or at least standard, in the treatment of a given topic"—"are not protectible [sic] by copyright"); *Landsberg v. Scrabble Crossword Game Players, Inc.*, 736 F.2d 485, 489 (9th Cir.) (*scènes à faire* not copyrightable because granting a copyright "would give the first author a monopoly on the commonplace ideas behind the *scènes à faire*"), cert. denied, 469 U.S. 1037, 105 S.Ct. 513, 83 L.Ed.2d 403 (1984).

If, however, the expression of an idea has elements that go beyond all functional elements of the idea itself, and beyond the obvious, and if there are numerous other ways of express-

ing the non-copyrightable idea, then those elements of expression, if original and substantial, are copyrightable.

### C. *Elements of the Legal Test for Copyrightability*

A "legal test," as I use the phrase here, is a statement of the elements of fact, or law, or both fact and law, that must be addressed by a decisionmaker to decide a question potentially decisive of some claim or defense. "Copyrightability" of non-literal elements of Lotus 1-2-3 is essential to the claim of the plaintiff in this case.

As already noted, the legal test for deciding copyrightability, in a factual context such as is presented here, has not been precisely defined either in the copyright statute or in precedents interpreting and applying it. Nevertheless, the statute and the precedents contain many mandates—"markers" of the borderline between copyrightability and non-copyrightability—that narrow the scope of the questions remaining to be answered to determine what test to apply.

Drawing into one statement the fundamental truths about ideas and their expression that were sketched above, one may accurately say that the issue of copyrightability of a "work" turns not on whether the work expresses ideas but instead on whether, in addition to expressing one or more ideas, in some material respect it does more, and in an original way. One need not totally disentangle the idea from its expression in order to conclude that a particular aspect is expression. Indeed, to speak as if it were ever possible completely to disentangle an idea from an expression of that idea is to speak abstract fiction rather than real-life fact. Disentanglement, then, is not an "either-or," "0-1," "negative-positive," or "binary" matter. It is, instead, a matter of degree.

Still, even if the "idea" cannot be completely disentangled from its expression, to determine what is copyrightable a decisionmaker must understand the meaning of "idea" within the idea-expression distinction. To do so one must take note also of another distinction—one between generality and specificity of conceptualizing the idea. Thus, a statement of

the most significant elements of the legal test for copyrightability, consistent with precedents, begins:

FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, *along the scale from the most generalized conception to the most particularized*, and choose some formulation—some conception or definition of the "idea"—for the purpose of distinguishing between the idea and its expression.

As Learned Hand recognized in a 1930 case concerning the alleged infringement of the copyright of a play:

Upon any work, and especially upon a play, a great number of patterns of increasing generality will fit equally ~~well~~; as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the play is about, and at times might consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his "ideas," to which, apart from their expression, his property is never extended. Nobody has ever been able to fix that boundary, and nobody ever can.

*Nichols*, 45 F.2d at 121 (citations omitted). See also *Shipman v. RKO, Radio Pictures*, 100 F.2d 533, 538 (2d Cir.1938) (L. Hand, J., concurring) ("*Nichols* . . . held that there is a point where the similarities are so little concrete (and therefore so abstract) that they become only 'theme', 'idea', or skeleton of the plot, and that these are always in the public domain; no copyright can protect them. The test is necessarily vague and nothing more definite can be said about it.") Thirty more years of experience in judging did not change Learned Hand's view: "Obviously, no principle can be stated as to when an imitator has gone beyond copying the 'idea,' and has borrowed its 'expression.' Decisions must therefore inevitably be ad hoc." *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir.1960) (L. Hand, J.). In another context,

Hand described such "ad hoc" decisionmaking as "fiat." *Sinram v. Pennsylvania Railroad Co.*, 61 F.2d 767, 771 (2d Cir.1932) (L. Hand, J.). In whatever way this kind of decisionmaking may be described, Hand offered us no formula for distinguishing between idea and expression like that he devised for the calculus of reasonable care in *United States v. Carroll Towing Co.*, 159 F.2d 169, 173 (L. Hand, J.) (articulating the "BPL" formula), *reh'g denied*, 160 F.2d 482 (2d Cir.1947). It seems the better part of wisdom, if not valor, not to press the search for a suitable brightline test of copyrightability where Learned Hand, even after decades of experience in judging, found none.

For all these reasons, as a practical necessity, whether explicitly or only implicitly, courts apply an abstractions scale in determining copyrightability. In doing so they make a decision involving choice and judgment of a type that human minds make regularly in daily affairs, but computers of the current state of the art cannot make.<sup>2</sup> Still, though "judgment" is required and the answer to be given is not precisely "calculable," analogies to arithmetic calculations and to "scales of justice" may aid the human mind in choosing and "weighing" factors that properly go into the metaphoric calculus.

In summary, one among the principal elements to be weighed in determining copyrightability when the idea-expression distinction applies is to conceive and define the idea in a way that places it somewhere along the scale of abstraction (somewhere between the most abstract and the most specific of all possible conceptions). Illustrations from the evidence in this case will help to explain in a more concrete way this element of the legal test for copyrightability. See Part V(C), *infra*. Before turning to those illustrations, however, I state in a similarly abstract way, to be explained

<sup>2</sup> Despite *Time Magazine's* decision to honor "the computer" as its 1982 "Man of the Year," and despite advances in the field of artificial intelligence, I take it as a premise of decisionmaking in this case that computers, for better or worse, do not yet have the human feelings, strengths, and failings that lie beyond those we describe as logical, or cognitive, or intellectual in the broadest sense.

later by illustrations, two more elements that I conclude a decisionmaker must consider to determine copyrightability of a computer program like that at issue in this case.

In addition to taking account of the distinction between generality and specificity, to make use of Hand's abstraction scale for applying the idea-expression distinction we need to identify and distinguish between essential and nonessential details of expressing the idea. Some, but of course not all, details, are so essential that their omission would result in a failure to express *that* idea, or in the expression of only a different and *more general* idea. Accordingly, two more elements in the legal test for copyrightability are:

SECOND, the decisionmaker must focus upon whether an alleged expression of the idea is limited to elements essential to expression of *that* idea (or is one of only a few ways of expressing the idea) or instead includes identifiable elements of expression not essential to every expression of that idea.

THIRD, having identified elements of expression not essential to every expression of the idea, the decisionmaker must focus on whether those elements are a substantial part of the allegedly copyrightable "work."

In addressing this third element of the test for copyrightability, the decisionmaker is measuring "substantiality" not merely on a quantum scale but by a test that is qualitative as well. *SAS Institute*, 605 F.Supp. at 829-30 ("the piracy of even a quantitatively small fragment ('a rose by any other name would smell as sweet') may be qualitatively substantial").

By its nature, a legal test that requires weighing of factors or elements such as these is not a bright-line or an either-or test. It requires of the decisionmaker, instead, an evaluative or "judgmental" weighing of all relevant characteristics of the work in which a copyright is claimed, all relevant characteristics of the allegedly infringing work, and all of the relevant circumstances of their development and use. It requires, also, not a step-by-step decisionmaking process, but a simultaneous weighing of all the factors or elements that the legal

test identifies as relevant. I do not suggest that the three elements identified here are an all-inclusive list. They do appear, however, to be the principal factors relevant to decision of copyrightability of a computer program such as Lotus 1-2-3.

If the decisionmaker, weighing the relevant factors, determines that the legal test applying the idea-expression distinction is satisfied, copyrightability is established. Issues may remain, of course, as to whether a copyright was perfected and whether the alleged infringing work, measured by the "substantial similarity" test, did contain elements that infringed upon the copyrightable elements of the copyrighted "work." Also, issues may remain as to whether damages have been proved, or whether the controversy over infringement is instead "a trivial pother, a mere point of honor, of scarcely more than irritation, involving no substantial interest," *Fred Fisher, Inc. v. Dillingham*, 298 F. 145, 152 (S.D.N.Y. 1924) (L. Hand, J.) (citations omitted), for which only statutory damages should be awarded. 17 U.S.C. § 504(c) (1988).

#### D. Incentives and the Role of Advocacy

The advocate of broad copyrightability who understands that the court will apply a legal test that focuses principally on these three elements has an incentive to urge that the court conceive the "idea" in a very generalized sense; then many different expressions of the idea would be possible, and protection might be claimed for the work. The advocate of freedom to copy—and of narrow copyrightability—has an incentive to urge that the court conceive the "idea" in a very particularized sense; then only one or a few expressions of an idea defined in such particularized terms would be possible, and no copyright protection for those few expressions would be available because the idea and expression would merge completely, or nearly so. Such extreme positions would fail to assist the court in determining, for purposes of the first element, where properly to place the idea along the abstractions scale.

There is risk for each advocate, however, in yielding too readily to these respective incentives. The argument of an

advocate who presses too far in one or the other of these directions of generality or specificity in defining the idea will not only lose the argument advanced but also lose credibility for later advancing a more sensible alternative that proposes a less extreme but still favorable position along the scale. Upon reflection, then, advocates on both sides will be—or at least should be—encouraged to moderate their ultra-contentious and extreme positions in favor of more supportable propositions that will more sharply focus the issue for adjudication.

There is an additional risk for the advocate of freedom to copy who attempts to persuade the court that Hand's abstractions scale, and the three-element test applying it, is irrelevant to computer programs. That advocate will be tempted to define the idea in the most generalized terms, hoping to persuade the court that, because the abstractions scale does not apply, all of the many different expressions of that idea are non-copyrightable and everyone is free to copy.

In this case, defendants have in fact advanced such an argument. Yielding only to the near-uniformity of precedent for copyrightability of source code and object code, defendants have argued that every other form of expression of the "spreadsheet metaphor" is non-copyrightable. For reasons now to be explained, neither this extreme contention nor any of the alternatives the defendants have advanced for non-copyrightability can be sustained.

## V. APPLICATION OF THE LEGAL TEST TO LOTUS 1-2-3

### A. "Look and Feel"

In musical, dramatic, and motion picture works, and works of literature, nonliteral elements that are copyrightable have sometimes been described as the "total concept and feel" of a work, *Roth Greeting Cards*, 429 F.2d at 1110; *Krofft Television*, 562 F.2d at 1167, "the fundamental essence or structure" of a work, 3 M. Nimmer & D. Nimmer, *Nimmer on Copyright* § 13.03[A][1] (1989), or "the 'pattern' of the

work," Chaffee, *Reflections on the Law of Copyright: Part I*, 45 Colum.L.Rev. 503, 513 (1945). In the context of computer programs, nonliteral elements have often been referred to as the "look and feel" of a program, e.g., *Telemarketing Resources*, 12 U.S.P.Q.2d at 1993; Samuelson and Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright "Look and Feel" Lawsuits*, 30 Jurimetrics 121 (1989). Initially, plaintiff too referred to these elements as "look and feel," Amended Complaint ¶ 13 (Docket No. 15), though plaintiff—in trial, at least—has not rested its contentions primarily on this terminology.

Despite its widespread use in public discourse on the copyrightability of nonliteral elements of computer programs, I have not found the "look and feel" concept, standing alone, to be significantly helpful in distinguishing between nonliteral elements of a computer program that are copyrightable and those that are not.

One may argue that the phrase "look and feel" is analogous to the "total concept and feel" test developed in *Roth Greeting Cards*, 429 F.2d at 1110, and used in *Krofft Television*, 562 F.2d at 1167. In these cases, however, the "total concept and feel" test was not invoked—at least, not explicitly—as an aid to the court in determining which nonliteral elements were copyrightable and why. Rather, these courts used the concept, not in determining copyrightability, but, apparently assuming copyrightability, in applying the substantial similarity test to determine whether forbidden copying had occurred. For example, in *Roth Greeting Cards*, the court considered whether the "text, arrangement of text, art work, and association between art work and text" of defendant's greeting cards were substantially similar to (e.g., copied the "total concept and feel" of) those elements of plaintiff's greeting cards. 429 F.2d at 1109. And in *Krofft Television*, the inquiry focused on whether defendant McDonald's, with its television commercials, copied the "total concept and feel" of plaintiff's H.R. Pufnstuf children's television series by copying its locale (both occurred in imaginary world with similar trees, caves, a pond, a road and a castle), fictional characters (both were

inhabited by talking trees, mayors with disproportionately large round heads and long wide mouths, "Keystone cops" characters, crazy scientists, and a multi-armed evil creature), costumes and sets (created for defendant by former employees of plaintiff), voices (executed for defendant by same voice expert who did the voices for plaintiff's television series), and plot. 562 F.2d at 1161, 1167 n. 9.

It may be true that the issues of copyrightability and substantial similarity are so interrelated that these precedents are relevant. The fact remains that they are not directly in point for determining copyrightability in this case. Moreover, "look and feel" is a conclusion, and the usefulness and applicability of a precedent depends on the *reasons* the conclusion was reached in a particular context, not on the *conclusion* itself. Thus, in trying to understand the relevance of "concept and feel" precedents, we need to look to details of those cases that appear to have been relied upon in reaching the conclusion, rather than merely embracing the conclusion without regard for underlying reasons. As we probe the circumstances of these precedents that closely, we are likely to do something akin to apply in the three-element test described above.

#### B. The User Interface

Plaintiff in the present case, not now pressing any argument that the phrase "look and feel" is a satisfactory description of the copyrightable elements of Lotus 1-2-3, suggests that the copyrightable nonliteral elements are more appropriately described by the phrase "user interface." According to plaintiff, the "user interface" of 1-2-3 includes such elements as "the menus (and their structure and organization), the long prompts, the screens on which they appear, the function key assignments, [and] the macro commands and language." Plaintiff's Post-Trial Brief at 53 (Docket No. 319). I turn now to examining these elements more closely.

Like manual spreadsheets, *Baker*, 101 U.S. at 100, the electronic spreadsheet presents a blank form on which numerical, statistical, financial or other data can be assimilated, organized, manipulated and calculated. Galler Decl. ¶¶ 101-103;

Lewis Affdvt. ¶ 92. In both Lotus 1-2-3 and VP-Planner, as in many other electronic spreadsheet programs, a highlighted element of the basic screen display resembles an "L" rotated ninety degrees clockwise with letters across the top to designate columns, and numbers down the left side to designate rows. See Appendices 1 (VisiCalc), 2 (1-2-3), and 3 (VP-Planner). Cf. Multiplan, which also has a rotated "L" screen display, but which uses numbers for both columns and rows (Tr. Ex. (Trial Exhibit) 132). The intersection of each column and row is a "cell" in which a value (e.g., 31,963), formula (e.g., one that adds a column of numbers), or label (e.g., "Cost of Goods") may be entered.

Both programs utilize a "two-line moving-cursor menu," which presents the user with a list of command choices (e.g., "file", "copy", "quit") and a moving cursor to use in communicating ("entering") the choice. The menu is called up to the screen by pressing the slash ("/") key, and is located either above the rotated "L" (as in 1-2-3, see Appendix 2) or below the rotated "L" as in VP-Planner, see Appendix 3). Cf. Multiplan, which uses a three-line moving cursor menu (Tr. Ex. 132); Excel, which has "pull-down" bar menus (Tr. Ex. 79).

The top line of the two-line menu contains a series of words, each of which represents a different command. For example, the top line of the first, or main, menu in 1-2-3 reads: "Worksheet Range Copy Move File Graph Data Quit". See Appendix 2; cf. Appendix 3 (main menu in VP-Planner). The first word of the line is highlighted to signify the command that will be chosen if the "enter" key is pressed; the highlighting, or "cursor," moves to the right or left if the right or left cursor key is pressed.

The second line of the menu displays a "long prompt," which contains further information about the highlighted command. In some cases, the long prompt is a description of the highlighted command (e.g., for command "Copy", the long prompt reads: "Copy a cell or range of cells"); in other cases, the long prompt provides a list of the menu command subchoices that will be available if the highlighted command is chosen (e.g., for command "Worksheet", the long prompt

reads: "Global, Insert, Delete, Column-Width, Erase, Titles, Window, Status"; see Appendix 2; cf. Appendix 3 (VP-Planner)). In the latter case, if the highlighted command is subsequently chosen, the words that appeared in the long prompt will now appear as second-level menu command choices on the top line of the menu, and a new long prompt will take its place on the second line.

In addition to having the option of selecting a command by moving the cursor to the command and pressing the "enter" key, a user may instead press the key representing the first letter of the command word (e.g., "C" for "Copy", "W" for "Worksheet"). For this reason, each word representing each command on a given menu line must start with a different letter. See *Galler Affdvt.* ¶¶ 105-117; *Lewis Decl.* ¶¶ 96-97.

Function keys present an additional way for the user to communicate with and operate, the programmed computer. Each program assigns certain frequently-used commands to the various "function keys" labelled "F1", "F2", "F3", etc.) on the keyboard. For example, in 1-2-3, "F1" corresponds to the command "Help", and "F2" to the command "Edit". *Galler Decl.* ¶ 125. VP-Planner, in contrast, assigns the function keys to the commands on the top line of the menu. Thus, for example, when VP-Planner is in its main menu mode, "F1" corresponds to the command "Help", "F2" to the command "Worksheet", "F3" to the command "Range", "F4" to the command "Copy", and so on. See Appendix 3 (function key numbers listed before command terms).

Typically, users adapt particular spreadsheets to their specific needs. Suppose, for example, that in order to achieve a desired result, a user must perform the same sequence of commands repeatedly in order to cause the computer to execute the same functions repeatedly (e.g., calculating depreciation based on certain financial data, or saving the spreadsheet and printing a copy of it). Rather than going step-by-step through the same sequence of commands each time there is a need to perform a particular function, the user may store a sequence of command terms as a "macroinstruction," commonly called a "macro," and then, with one command stroke that invokes

the macro, cause the programmed computer to execute the entire sequence of commands.

In 1-2-3, the command terms within a macro may consist of menu choices (e.g., "/C" to copy a cell or range of cells, or "/PPRA1.F19[enter]AGQ"—for the command sequence "Print, Printer, Range, Cell A1 to Cell F19, Align, Go, Quit"—to print the specified range of cells from the spreadsheet). Also a macro may include keyboard commands (such as function keys, cursor keys, or the "enter" key), and special macro commands invoked by the command "/X" (e.g., "XI" is a macro command that performs an "if-then" logical function). Because macros may contain many menu choices, the exact hierarchy—or structure, sequence and organization—of the menu system is a fundamental part of the functionality of the macros. Also, because macro commands are typically invoked by entering the first letter of command choices (e.g., "/C" for copying, "/PPRA1.F19[enter]AGQ" for printing the specified range of cells), the first letter of each command choice on a particular menu is a vital element of the functionality of macros. See *Galler Decl.* ¶¶ 121-124; *Lewis Affdvt.* ¶¶ 109-115.

### C. *Elements of the User Interface as Expression*

Applying to 1-2-3 the legal test stated in Part IV(C), *supra*, I consider first where along the scale of abstraction to conceive the "idea" for the purpose of distinguishing between the idea and its expression.

At the most general level of Hand's abstractions scale, *Nichols*, 45 F.2d at 121—the computer programs at issue in this case, and other computer programs that have been considered during the course of trial, are expressions of the idea of a computer program for an electronic spreadsheet. Defendants are quite correct, then, in asserting that the idea of developing an electronic spreadsheet is not copyrightable—that the core idea of such a spreadsheet is both functional and obvious, even to computer users who claim no technical competence. Thus, even though programs like VisiCalc, 1-2-3, Multiplan, SuperCalc4, and Excel are very different in their

structure, appearance, and method of operation, each is, at the most basic level, just a different way of expressing the same idea: the electronic spreadsheet. It does not follow, however, that every possible method of designing a metaphorical spreadsheet is obvious, or that no form of expressing the idea of the spreadsheet metaphor can possibly have such originality in pressing beyond the obvious as is required for copyrightability, or that no special form of metaphorical spreadsheet can possibly be a distinctive expression of a particular method of preparing financial information.

The idea for a two-line moving cursor menu is also functional and obvious, and indeed, is used in a wide variety of computer programs including spreadsheet programs. Nevertheless, it does not follow that every possible method of designing a menu system that includes a two-line moving cursor is non-copyrightable.

Of course, if a particular expression of the idea of an electronic spreadsheet communicates no details beyond those essential to stating the idea itself, then that expression would not be copyrightable. The issue here is whether Lotus 1-2-3 does go beyond those details essential to any expression of the idea, and includes substantial elements of expression, distinctive and original, which are thus copyrightable.

The idea for an electronic spreadsheet was first rendered into commercial practice by Daniel Bricklin. As a student at Harvard Business School in the late 1970's, Bricklin envisioned a "magic blackboard" that would recalculate numbers automatically as changes were made in other parts of the spreadsheet. Eventually, aided by others, he transformed this idea into VisiCalc, the first commercial electronic spreadsheet. *See Bricklin Affdvt.* ¶ 48-96 (Docket No. 217). Bricklin's idea for VisiCalc was a revolutionary advance in the field of computer programming. *Dauphinais Affdvt.* ¶ 98 (Docket No. 280).

Although VisiCalc was a commercial success, implementational characteristics limited the scope and duration of its marketability as a spreadsheet product. Most notably, VisiCalc was originally programmed for use on the Apple II computer, which had limited memory (32K of RAM), limited

screen display capabilities (only 40 characters per line), and limited keys available on the keyboard (no function keys and no up and down cursor keys). When VisiCalc was later rewritten for use on the IBM PC which was introduced in August 1981), it was transferred with minimal changes and without taking advantage of many of the PC's more extensive capabilities.

Mitchell Kapor and Jonathan Sachs, the original authors of 1-2-3, exploited this opportunity. Building on Bricklin's revolutionary idea for an electronic spreadsheet, Kapor and Sachs expressed that idea in a different, more powerful way. 1-2-3 took advantage of the IBM PC's more expansive memory and more versatile screen display capabilities and keyboard. 1-2-3, like many electronic spreadsheet programs since, could thus be thought of as an evolutionary product that was built upon the shoulders of VisiCalc.

Just as 1-2-3 expressed the idea of an electronic spreadsheet differently from VisiCalc, so did Microsoft's Excel. Originally written for the Apple Macintosh computer, it exploits the enhanced graphics capabilities of the Macintosh, as well as the mouse input device that is standard with the Macintosh. Excel has pull-down bar menus rather than a two-line moving-cursor menu, and a very different menu-command hierarchy. *Tr. Ex. 79.*

As already noted, these three products—VisiCalc, 1-2-3, and Excel—share the general idea of an electronic spreadsheet but have expressed the idea in substantially different ways. These products also share some elements, however, at a somewhat more detailed or specific point along the abstractions scale. One element shared by these and many other programs is the basic spreadsheet screen display that resembles a rotated "L." *See Appendices 1 (VisiCalc), 2 (1-2-3), and 3 (VP-Planner).* Although Excel uses a different basic spreadsheet screen display that more closely resembles a paper spreadsheet, there is a rather low limit, as a factual matter, on the number of ways of making a computer screen resemble a spreadsheet. Accordingly, this aspect of electronic spreadsheet computer programs, if not present in every expression of such a program, is present in most expressions. Thus the

second element of the legal test weighs heavily against treating the rotated "L" screen display as a copyrightable element of a computer program. *Morrissey*, 379 F.2d at 678-79.

Another expressive element that merges with the idea of an electronic spreadsheet—that is, that is an essential detail present in most if not all expressions of an electronic spreadsheet—is the designation of a particular key that, when pressed, will invoke the menu command system. The number of keys available for this designation is limited for two reasons. First, because most of the keys on the keyboard relate either to values (*e.g.*, the number keys and mathematical operation keys) or labels (*e.g.*, the letter keys), only a few keys are left that can be used, as a practical matter, to invoke the menu command system. Without something more, the programmed computer would interpret the activation of one of these keys as an attempt by the user to enter a value or label into a cell. Second, because users need to invoke the command system frequently, the key designated for this purpose must be easily accessible. For example, the user should not be required to press two keys at the same time (such as "Shift," "Alt," or "Ctrl" along with another key).

As just noted, when all the letter, number, and arithmetic keys are eliminated from consideration, the number of keys remaining that could be used to invoke the menu command system is quite limited. They include the slash key ("/") and the semi-colon key (";"). The choice of the creators of VisiCalc to designate the slash ("/") key to invoke the menu command system is not surprising. It is one of very few practical options. Thus the second element of the legal test weighs heavily against copyrightability of this aspect of VisiCalc—and of 1-2-3. This expression merges with the idea of having a readily available method of invoking the menu command system.

Other elements of expression a decisionmaker may regard as either essential to every expression of an electronic spreadsheet, or at least "obvious" if not essential, include the use of the "+" key to indicate addition, the "-" key to indicate subtraction, the "\*" key to indicate multiplication, the "/" key within for-

mulas to indicate division, and the "enter" key to place keystroke entries into the cells. *See Dauphinais Affidavit*, ¶ 78.

Each of the elements just described is present in, if not all, at least most expressions of an electronic spreadsheet computer program. Other aspects of these programs, however, need not be present in every expression of an electronic spreadsheet. An example of distinctive details of expression is the precise "structure, sequence, and organization," *Whelan*, 797 F.2d at 1248, of the menu command system.

Consider first the menu command system of VisiCalc. The main menu command line reads: "Command: BCDEFGIM-PRSTVW-". *See* Appendix 1. Each of these letters (or, to use defendants' experts' preferred terminology, "symbolic tokens") stands for a different command—in this case: Blank, Clear, Delete, Edit, Format, Global, Insert, Move, Print, Replicate, Storage, Titles, Version Number, Window, and "-" for "Label Repeating." Many of these commands invoke submenus which also contain a series of letters, each of which represents a submenu command choice. *See* VisiCalc Command Structure Chart (Tr. Ex. 140, pp. 3-3 and 3-4).

This particular expression of a menu structure is not essential to the electronic spreadsheet idea, nor does it merge with the somewhat less abstract idea of a menu structure for an electronic spreadsheet. The idea of a menu structure—including the overall structure, the order of commands in each menu line, the choice of letters, words, or "symbolic tokens" to represent each command, the presentation of these symbolic tokens on the screen (*i.e.*, first letter only, abbreviations, full words, full words with one or more letters capitalized or underlined), the type of menu system used (*i.e.*, one-, two-, or three-line moving-cursor menus, pull-down menus, or command-driven interfaces), and the long prompts—could be expressed in a great many if not literally unlimited number of ways.

The fact that some of these specific command terms are quite obvious or merge with the idea of such a particular command term does not preclude copyrightability for the command structure taken as a whole. If particular characteristics

not distinctive individually have been brought together in a way that makes the "whole" a distinctive expression of an idea—one of many possible ways of expressing it—then the "whole" may be copyrightable. The statutory provisions regarding "compilation," 17 U.S.C. §§ 101, 103, are not essential to this conclusion, but do reinforce it. A different total structure may be developed even from individual components that are quite similar and limited in number. To determine copyrightability, a court need not—and, indeed, should not—dissect every element of the allegedly protected work. Rather, the court need only identify those elements that are copyrightable, and then determine whether those elements, *considered as a whole*, have been impermissibly copied. *Atari Games Corp. v. Oman*, 888 F.2d 878, 882-83 (D.C.Cir. 1989) (rejecting "component-by-component analysis," and ruling instead that focus must ultimately be on "work as a whole").

It is plain that plaintiff did not impermissibly copy copyrighted elements of VisiCalc. Lotus 1-2-3 uses a very different menu structure. In contrast with VisiCalc's one-line main menu that reads "Command: BCDEFGIMPRSTVW-", the main menu of Lotus 1-2-3, which uses a two-line moving-cursor menu system, reads: "Worksheet Range Copy Move File Graph Data Quit". See Appendix 2. Most of the submenus similarly present a list of up to about ten full-word menu choices, presented in order of predicted frequency of use rather than alphabetically. See Lotus 1-2-3 Command Tree Chart (Tr. Ex. 176, pp. 3-2 and 3-3). Other spreadsheet programs have also expressed their command structures in completely different ways. Cf. Multiplan (Tr. Ex. 132, p. 279-374), Framework II (Tr. Ex. 131, pp. QR7-QR8), SuperCalc4 (Tr. Ex. 133, Quick Reference Card Number 1), Excel (Tr. Ex. 79, Reference Guide, p. 90); Symphony (Tr. Ex. 130, Reference Manual, pp. 127-190); MathPlan (Tr. Ex. 134, pp. 169-380); and PFS: Professional Plan (Tr. Ex. 135).

I conclude that a menu command structure is capable of being expressed in many if not an unlimited number of ways, and that the command structure of 1-2-3 is an original and nonobvious way of expressing a command structure. Emery

Decl. ¶ 15. Accordingly, the menu structure, taken as a whole—including the choice of command terms, the structure and order of those terms, their presentation on the screen, and the long prompts—is an aspect of 1-2-3 that is not present in every expression of an electronic spreadsheet. It meets the requirements of the second element of the legal test for copyrightability.

Finally, I consider the third element of the legal test—whether the structure, sequence, and organization of the menu command system is a substantial part of the alleged copyrighted work—here Lotus 1-2-3. That the answer to this question is "yes" is incontrovertible. The user interface of 1-2-3 is its most unique element, and is the aspect that has made 1-2-3 so popular. That defendants went to such trouble to copy that element is a testament to its substantiality. Accordingly, evaluation of the third element of the legal test weighs heavily in favor of Lotus.

Taking account of all three elements of the legal test, I determine that copyrightability of the user interface of 1-2-3 is established.

## VI. COPYING OF LOTUS 1-2-3

As noted at the beginning of this Opinion, the parties' stipulation regulating this first phase of trial reserved for a later, jury phase, any issues of fact requiring jury determination with respect to any alleged copying of Lotus 1-2-3. If in this first phase the court had rejected Lotus' claim of copyrightability of nonliteral elements of the user interface, a jury phase would plainly have been required under that stipulation to determine whether any copying of source code or object code had occurred. Because, however, the court has decided instead that Lotus prevails on this issue, the court must next consider whether any issue remains that must be submitted to a jury before the court considers whether defendants are liable for infringement of Lotus' copyright in 1-2-3.

For the reasons stated below, the answer to that inquiry must be "no." Not only is the copying in this case so "overwhelming and pervasive" as to preclude, as a matter of law, any assertion of independent creation, *see Midway Manufacturing Co. v. Bandai-America, Inc.*, 546 F.Supp. 125, 141 n. 11, 149 (D.N.J.1982) ("overwhelming and pervasive" copying can preclude, as a matter of law, finding of independent creation, and can support grant of summary judgment for plaintiff, but such virtual identity not shown in this case), but also, defendants in this case have *admitted* that they copied these elements of protected expression.

Dr. James Stephenson, founder of defendant Stephenson Software, is the original developer of the program that was eventually released as VP-Planner. Like Kapur and Sacks, Stephenson recognized that VisiCalc, although a "pioneering spreadsheet approach," Stephenson Affdvt. ¶ 19 (Docket No. 287), was not sufficient to meet the financial planning needs of some companies and did not take advantage of technological advances in computer hardware. Accordingly, in January 1982, Stephenson began to develop his own electronic spreadsheet that he referred to as FIPS ("Financial Information and Planning System"). *See id.* at ¶¶ 13-59.

By January 1983, Stephenson had developed much of the user interface for his spreadsheet program including the menu command hierarchy. *See Tr. Ex. 1014.* By April 1983, Stephenson installed at his client's business an operational version of his spreadsheet program, which had substantially the same menu hierarchy as the January 1983 version. That hierarchy is differently expressed from the hierarchy of both VisiCalc and 1-2-3. *See Tr. Ex. 1019; Stephenson Affdvt. ¶¶ 56-60.*

Stephenson first saw 1-2-3 in operation in February 1983, after he had developed the menu hierarchy for FIPS. Throughout the rest of that year, he continued to improve FIPS, changed its name VP-Planner, and began to consider marketing his program. By December 1983, Stephenson entered into a letter of intent with Adam Osborne regarding publication of VP-Planner. Osborne thereafter organized defendant Paperback Software. Stephenson Affdvt. ¶¶ 66-82.

Throughout 1984, defendants continued to improve VP-Planner. In the autumn, they recognized the success of 1-2-3 and reached the conclusion that spawned this litigation: VP-Planner, in order to be a commercial success, would have to be "compatible" with 1-2-3. "The only way to accomplish this result," defendants believed, "was to ensure that *the arrangement and names of commands and menus in VP-Planner conformed to that of Lotus 1-2-3.*" *Id.* at ¶ 117(emphasis added). *See generally id.* at ¶¶ 99-130. Such compatibility would allow users to transfer spreadsheets created in 1-2-3 to VP-Planner without loss of functionality for any macros in the spreadsheet. Also, such compatibility would allow users to switch from 1-2-3 to VP-Planner without requiring retraining in the operation of VP-Planner.

To some degree at least, defendants' premises have proved incorrect in hindsight. That is, first, as Excel has proved, a spreadsheet program did not have to be exactly compatible with 1-2-3 in order to be a commercial success. Second, copying the menu structure was not the *only* way to achieve aspects of this desired compatibility. For example, defendants could have instead added a macro conversion capability as the creators of Excel have successfully done (the Microsoft Excel Macro Translation Assistant), and could have provided an on-line help function that would show users the VP-Planner equivalent for 1-2-3 commands. *See Excel Reference Guide* at 491, 425-26 (Tr. Ex. 79). *See also Morgan Decl. ¶¶ 3-7* (Docket No. 308) (Lotus itself created a "macro conversion utility" to translate macros among different-language editions of 1-2-3 (e.g., North American, international English, French, German, Italian, Spanish, and Swedish)); *Turner Decl. ¶ 10* (Docket No. 309). These points do not weigh significantly in the present decision, however, because even if VP-Planner otherwise would have been a commercial failure, and even if no other technological ways of achieving macro and menu compatibility existed, the desire to achieve "compatibility" or "standardization" cannot override the rights of authors to a limited monopoly in the expression embodied in their intellectual "work."

Defendants admit that, once these fateful decisions were made by Stephenson and Osborne, defendants converted VP-Planner into a program more like 1-2-3—indeed, a program that they have publicly advertised as a “workalike for 1-2-3.” *VP Planner Manual* 1.11 (1985) (Tr. Ex. 9). It is incontrovertible that, in the process, they copied the expressive elements of 1-2-3 that the court has concluded are copyrightable:

[M]aking the changes required for macro compatibility meant that we had to revise existing elements of the [VP-Planner] spreadsheet interface, including the hierarchical menu structure; ensure that keystroke sequences would bring about the same operational result in both programs; add certain functional elements found in Lotus 1-2-3 which VP-Planner did not yet support; and discard certain features which, although beneficial, were inconsistent with the macro compatibility requirement . . . .

Several types of changes were required in the VP-Planner program to achieve keystroke macro compatibility. First, the menu structure had to be altered so that all menu commands would have the same first letter and be in the same location in the menu hierarchy as in Lotus 1-2-3.

Stephenson Affdvt. ¶¶ 144, 146. *See generally id.* at ¶¶ 142-157. After these changes were made, the VP-Planner manual could truthfully declare:

VP-Planner is designed to work like Lotus 1-2-3, keystroke for keystroke. . . . VP-Planner’s worksheet is a feature-for-feature workalike for 1-2-3. It does macros. It has the same command tree. It allows the same kind of calculations, the same kind of numerical information. Everything 1-2-3 does, VP-Planner does.

*VP-Planner Manual* at xi. 1.11.

The court’s comparison of the 1-2-3 menu command hierarchy and the VP-Planner menu hierarchy confirms that VP-Planner “has the same command tree” as 1-2-3—that is, that defendants copied the expression embodied in the 1-2-3 menu

hierarchy. It is true that there are some differences between 1-2-3’s menu structure and VP-Planner’s menu structure. For example, most VP-Planner menu lines begin with a help (“?”) command, and some additional commands are included at the end of some menu lines (*i.e.*, “DBase, Multidimensional” on the “/File Erase” menu line; and “Page #, No Page #, Row/Col. #, Stop Row/Col. #, Background” on the “/Print Printer Options Other” menu line). Other differences between the two programs appear in the start-up screens, the placement on the screen of the menu lines, the exact wording of the long prompts, the organization of the help screens, the increased width of the VP-Planner screen, and the ability of VP-Planner to hide certain columns. *See Stephenson Affdvt.* ¶¶ 161-176. The works are, nevertheless, substantially, indeed, strikingly, similar. As Judge Learned Hand held in a copyright case involving a pattern on a bolt of cloth that was used to make dresses, infringement may be found despite some differences between two works:

the ordinary observer, unless he set out to detect the disparities, would be disposed to overlook them, and regard their aesthetic appeal as the same. That is enough; and indeed, it is all that can be said, unless protection against infringement is to be denied because of variants irrelevant to the purpose for which the design is intended.

*Peter Pan Fabrics*, 274 F.2d at 489, *quoted in part with approval*, *Concrete Machinery*, 843 F.2d at 607. *See also Atari v. Philips*, 672 F.2d at 618 (“[A] laundry list of specific differences . . . will not preclude a finding of infringement where the works are substantially similar in other respects. . . . When analyzing two works to determine whether they are substantially similar, courts should be careful not to lose sight of the forest for the trees.”).

From the perspective of both an expert and an ordinary viewer, the similarities overwhelm differences. Thus, as in *Peter Pan Fabrics*, the two works at issue are substantially similar. Indeed, by using the option in VP-Planner that allows a user to move the menu from the bottom of the screen to the

top of the screen' a user could easily think 1-2-3 rather than VP-Planner was the program in use. Certainly purchasers of a book designed to teach users how to master 1-2-3, which is distributed with demonstration copies of *VP-Planner*, would be likely to overlook the disparities between 1-2-3 and VP-Planner. See L. Ingalsbe, *Lotus 1-2-3 with Version 2.0 for the IBM PC* (1987) (Tr. Ex. 215).

Moreover, even if some elements of VP-Planner were very different, it would not give defendants a license to copy other substantial elements of 1-2-3 verbatim. If one publishes a 1,000-page book of which only a 10-page segment is an unauthorized reproduction of copyrighted material, and if the 10-page segment is a qualitatively substantial part of the copyrighted work, it is not a defense to a claim of infringement that the book is 99% different from the copyrighted material. *Sheldon*, 81 F.2d at 56 ("no plagiarist can excuse the wrong by showing how much of his work he did not pirate"); *SAS Institute*, 605 F.Supp. at 829-30. Thus, defendants' proof that VP-Planner has many features that are different from Lotus 1-2-3 is off point. The more relevant question is: does it have significant features that are substantially similar? I conclude, on the record before me, that there is no genuine dispute of material fact on this question. The answer to this question must be "yes."

Accordingly, I conclude that it is indisputable that defendants have copied substantial copyrightable elements of plaintiff's copyrighted work. I therefore conclude that, subject to consideration of other contentions advanced by defendants, liability has been established.

## VII. A POSTSCRIPT ON THE NATURE OF DECISIONMAKING IN THIS CASE

### A. Policy Arguments and Limitations on the Role of Courts

Although Parts I through VI of this Opinion are sufficient to demonstrate that defendants have impermissibly copied copyrightable elements of expression embodied in 1-2-3, it may be useful to pause, before reaching the merits of defen-

dants' proffered defenses to liability, to consider the proper use of policy in determining statutory meaning.

As noted in Part III(A)(1), *supra*, the court is directed to look to the "object and policy" of the law in determining the manifested meaning of the law. *Kelly*, 479 U.S. at 43, 107 S.Ct. at 358. The court must, however, be careful that its weighing of the "object and policy" implicit in the law does not override the mandates that are either explicit in the statutory text or implicit in the provisions of the whole law.

Defendants' arguments in this case have urged the court to violate that limitation on the judicial use of policy considerations. That is, in the name of the "object and policy" of copyright law, defendants have argued: (1) that the idea-expression distinction creates unnecessary and counter-productive uncertainty that could be avoided by a bright-line test of copyrightability; (2) that the policy of promoting creativity favors extremely narrow copyright protection in computer programs for anything beyond source code and object code; and (3) that the policy of encouraging standardization mandates the conclusion that a user interface is not copyrightable. Also, in advancing these policy arguments, defendants have obscured the nature of important issues in the case by strained analogies and word games.

Policy arguments advanced without focusing on whether they are inconsistent with the mandates of legislation—along with strained analogies and word games—are fundamentally flawed because they do not take account of limitations on the role of courts, to which counsel as well as courts are obliged to be sensitive. Courts must enforce the mandates of legislation, and even when answering questions of law not answered by those mandates, must do so in a way that respects legislative policy choices, whether explicit or implicit.

Each of defendants' policy arguments addressed below suffers from this fatal flaw.

### B. Strained Analogies and Word Games

The reasoning that lawyers propose in arguments and judges use in deciding cases often builds on simile, metaphor,

and allusion. All such reasoning is within the scope of what we call analogy, in the broadest sense. When strictly logical deductions from authoritative declarations of law—in constitutions, statutes, and precedents—do not provide answers to questions of law that must be decided in the case before the court, the court turns to analogy. *Cf. Kelly*, 479 U.S. at 43, 107 S.Ct. at 358 (“we must . . . look to the provisions of the whole law”). Thus, similes, metaphors, and other forms of allusion are appropriately a part of our efforts to communicate ideas. They are not to be mistaken, however, for logically compelled inferences from authoritative declarations. They are aids to understanding the meaning, not yet authoritatively determined, of the authoritative declarations as applied to circumstances those declarations did not explicitly address.

Defendants strongly urged the court to accept an analogy between a computer program and a “useful article.” As explained in Part IV(A) of this Opinion, that proposed analogy is deeply flawed because of an unstated implicit premise that the fact that a computer program is “useful” utterly precludes copyrightability of any expressive elements in it. In the same breath, though, defendants concede, that source code and object code are copyrightable. This concession is itself a demonstration of the vulnerability of the “useful article” argument. The concession makes clear a flaw in the analogical argument for treating a computer program that is “useful” as necessarily nothing more than a “useful article.” If no explanation is offered for the exception other than precedent, the analogy itself is challenged by the precedent.

Thus, labeling a computer program as a “useful article” in relation to a claim for copyrightability of nonliteral elements of the user interface, but not a “useful article” in relation to a claim for copyrightability of source code and object code, is engaging in a word game rather than focusing on the merits of the copyrightability issue.

An even more striking word-game argument is defendants contention that in copying the 1-2-3 user interface, they have only copied a “language,” and that languages are not copyrightable. *See, e.g.,* Deposition of Steven Cook at 153:23-

154:10 (Docket No. 327); Kocher Affdvt. ¶¶ 13-18 (Docket No. 272); Defendants’ Post-Trial Brief at 87-103 (Docket No. 325). Although defendants have not explicitly stated all the essential steps of this argument, by filling the gaps in trying to understand the argument, one may infer the following elements and corollaries of the argument:

(1) Although expression is copyrightable, the language in which the expression is written is not copyrightable. Thus, a book written in English or French may be copyrightable, but the English and French “languages” are not works in which copyright may subsist.

(2) Like books, computer programs, written in computer programming “languages,” may be copyrightable, but only the “sets of statements or instructions,” and not the “language” in which they are written, are copyrightable.

(3) 1-2-3 has a macro capability or facility that allows a user to “write” her or his own macros. When the user writes a macro, he or she is actually writing a kind of computer “program.” *See* 1-2-3 Release 1A User’s Manual at 117 (macro “facility gives the 1-2-3 user a true programming capability”) (Tr. Ex. 72); 1-2-3 Release 2.01 User’s Manual at 3-10 (“1-2-3 includes a number of macro commands that create a powerful programming language”) (Tr. Ex. 72B).

(5) The macro, or “program,” that the user writes may be copyrightable if original and nonobvious. The “language” in which the macro is written is never copyrightable.

(6) Thus, when defendants copied the menu command hierarchy from 1-2-3, they did not copy a copyrightable element that embodied expression, but rather, copied only the “macro facility language” of 1-2-3, a non-copyrightable element in the public domain.

The vulnerable steps of this argument include, at a minimum:

(1) That “language” has a single, invariable meaning in all discourse about “languages”;

(2) That not only languages such as English and French but all other languages as well—including Esperanto, and *Reiss*' coined words, 276 F. at 718, and Pascal—are automatically ineligible for copyright protection (as to vulnerability of this step of the argument, defendants, though invited at trial to do so, have cited no precedent that supports the contention that a "language," even if original, is not copyrightable);

(3) That "language" and "sets of statements or instructions" are opposites, and never the twain shall meet—that is, anything that is a "set of statements or instructions" is not a "programming language," and anything that is a "programming language" is not "a set of instructions."

Just to state these implicit elements of defendants' argument is sufficient to demonstrate that the argument depends on arbitrary definitions of words, adopted for undisclosed reasons. In human experience, words have variable meanings; dictionaries seldom list a single definition for a word. An argument that depends on the proponents' undisclosed definitions of words—and even different definitions as a word is used in different steps of the argument—becomes a word game that obscures the substantive meaning of the argument and is an obstruction rather than an aid to the court's use of the adversary process to inform and thus improve decision-making.

Having explored the argument fully to try to understand its true nature, I conclude that defendants' "language" argument about the macro facility of Lotus 1-2-3, like defendants' "useful article" argument examined in Part IV(a), *infra*, is totally without merit.

### C. Policy Arguments for Bright-Line Rules

One of the themes in defendants' bundle of policy arguments is a plea for bright-line certainty. Indeed, it is a persistent theme throughout defendants' submissions: Software developers would like to know what they may and what they must not copy. The idea-expression distinction, defendants argue, fails to achieve this policy goal; therefore, the court

should adopt instead the bright-line literal-nonliteral distinction.

The yearning for certainty is not unique to this case or to copyright law. But certainty has a price. It reflects a hope that the court will apply hard-and-fast rules despite circumstances that cry out for a judgmental standard of decisionmaking that takes account of circumstances an impartial observer would think relevant to a "just" disposition of the case—which, after all, is one of the declared objectives of the legal system. *E.g.* Fed.R.Civ.P. 1 ("just, speedy, and inexpensive determination of every action").

The more deeply one probes, whether as lawyer or judge, or as an interested observer or critic of the legal system, the clearer it becomes that in many circumstances hard-and-fast rules, despite their initial attractiveness and false promise of certainty, have consequences that offend one's sense of justice. For example, hard-and-fast rules, because they allow for little or no evaluation and discretion, sometimes fail to take account of the competing values underlying the relevant law, and fail even to attempt to find an accommodation that serves conflicting high-value interests as well as possible, and at the lowest possible detriment to each. Professionals in law are forced by the nature of their work to do more probing into the disadvantages of bright-line rules than other observers generally are likely to do. For this reason, professionals are less likely than other observers to expect decisionmaking based on hard-and-fast rules that, in hindsight, seems arbitrarily to disregard relevant circumstances and to deserve the pejorative label "legalistic."

In this case, defendants have nevertheless proposed a bright-line rule that would extend copyright only to source code and object code, and never to nonliteral elements of computer programs. That proposal must be rejected. Not only is the proposed distinction contrary to the statutory mandates discussed throughout this Opinion, but also, it reflects the false hopes that are endemic to trying to apply brightline rules where judgmental, evaluative standards are more appropriate. *Nichols*, 45 F.2d at 121; *Peter Pan Fabrics*, 274 F.2d at 489.

In any event, as noted in Part II, Congress could have constitutionally drawn defendants' proposed bright-line—providing protection only to computer program code and not to nonliteral elements of computer programs—but Congress has not done so. Without a congressional mandate, it would be an abuse of authority for this court, in deciding this case, to use a bright-line test of copyrightability that makes the literal-nonliteral distinction decisive. Instead, the court must adjudicate within the area of protection mandated by Congress.

#### D. *Opinion Evidence and Premises of Legal Rulings*

Defendants, though disavowing any intent to invite the court to engage in lawmaking, have gone to extraordinary lengths in this case to advance policy arguments that a court can appropriately consider only if, in order to decide the case *sub judice*, it must answer previously unanswered questions about the meaning of relevant constitutional, statutory, and judicial sources of authority. Whether such a determination of previously undetermined meaning of an authoritative declaration is described as lawmaking, as I believe it to be, or instead in some other way—for example, as determining what “old statutes have since become” in circumstances materially different from those existing when the statute was enacted, R. Dworkin, *Law's Empire*, 248-50 (1986)—is of less consequence than how a court goes about deciding meaning for the case at hand. What “facts” or “evidence” shall the court consider? Shall the court consider expert opinions about what the circumstances were at the time the statute was enacted, and how they differ now?

Defendants' submissions have failed to address directly these basic questions about the appropriate role of courts. Implicitly, however, they have asked the court to consider policy arguments as freely as if the court were under no constraints in this respect. Defendants, for example, offered the following opinion testimony of a witness qualified as an expert in computer software development:

If the law is interpreted to say that a developer may not make a new program that can use the data files and

inputs accepted by existing programs, then computer advancement as we know it will be slowed. . . .

I believe that such a decision would throw the entire software industry into confusion. . . . [T]here would be a chilling effect on development and advancement in many areas. . . .

This case has implications about whether or not new, improved programs can read and execute data users created for themselves with an older program. The quantity of data to be affected by this decision is incomprehensively vast. It is not a case about “cheap copies”.

If aspects of screen displays that are governed by functionality (such as a command language like the 1-2-3 command structure) are held to be within the scope of copyright protection, then progress in application and systems computer programs could be dramatically slowed in the United States.

Bricklin Affdvt. ¶¶ 176-77, 179-80. *See also* Samuelson & Glushko, *Comparing the Views of Lawyers and User Interface Designers*, 30 *Jurimetrics* at 137 (concluding on basis of survey that leading user interface designers oppose strong copyright protection for elements of a user interface because such protection would be harmful, rather than helpful, to the industry).

By rulings during trial, the court received opinion evidence of this kind into the “record” in the broadest sense, even though excluding it from consideration in relation to any material adjudicative fact that might be in dispute. Not surprisingly, defendants, though explicitly invited to do so, were never able to identify any “adjudicative fact” question, as that term is used in Rule 201 of the Federal Rules of Evidence, to which opinion evidence such as this would be relevant. The court nevertheless received this evidence, over plaintiff's objection on the ground of irrelevance, because courts are free to consider “evidence” of this type when deciding “legislative” (or “premise”) facts. That is, courts may consider

such evidence to resolve disputable assertions of an evaluative nature (either about the past or about the future) or assertions about historical facts, or predictions about future consequences of adopting one or another rule of law, as premises for deciding what legal ruling to make. Cf.: F.R.E. 201(a), Notes of Advisory Committee (explaining "legislative facts"). See also *In re Asbestos Litigation*, 829 F.2d 1233, 1245-52 (3d Cir.1987) (concurring opinion of Becker, J.), cert. denied, 485 U.S. 1029, 108 S.Ct. 1586, 99 L.Ed.2d 901 (1988). This is not to say, however, that either the trial court that allows the evidence to be received into the record or an appellate court on review is limited by the rules of admissibility and the standards of review that apply to disputed issues of "adjudicative" fact. *Chastleton Corp. v. Sinclair*, 264 U.S. 543, 548, 44 S.Ct. 405, 406, 68 L.Ed. 841 (1924) ("the Court may ascertain as it sees fit any fact that is merely a ground for laying down a rule of law"). Cf. *Lockhart v. McCree*, 476 U.S. 162, 168-69 n. 3, 106 S.Ct. 1758, 1762-63 n. 3, 90 L.Ed.2d 137 (1986) ("We are far from persuaded, however, that the 'clearly erroneous' standard of Rule 52(a) applies to the kind of 'legislative' facts at issue here."). See also *Dunagin v. City of Oxford, Mississippi*, 718 F.2d 738, 748 n. 8 (5th Cir.1983) (en banc) ("legislative" fact discussion by Reavley, J.) ("If the legislative decision is not binding at this stage, at least it carries great weight. Certainly it cannot be thrust aside by two experts [who were called as witnesses before the trial court] and a judicial trier of fact."), cert. denied, 467 U.S. 1259, 104 S.Ct. 3554, 82 L.Ed.2d 855 (1984).

The court having received defendants' proffered evidence into the record for the limited purpose of considering its weight, if any, as bearing on premises of any legal ruling the court might find it necessary to make in deciding the case, plaintiff responded with sharply contrasting opinions in its post-trial brief:

*First*, the tremendous growth and success of the U.S. software industry is the direct result of the creative and original efforts of its software developers, laboring under

the protection of the copyright laws. Innovation has been the key to market success. . . .

*Second*, to the extent that defendants attempt to characterize this struggle as pitting large heartless corporate giants against lonely and defenseless developers working out of their dens and basements, the defendants do not understand what is at stake here. If the elements of a computer program at issue here were to lose their copyright protection, the biggest losers of all would be the small developers. The history of this industry has been one of creative designers who identify an unfilled need in the market and then design and build a superior product to fill that need. . . . [T]he developers' ability to realize substantial rewards for their creative efforts has depended entirely upon the legal protection copyright has afforded their work.

If the defendants could rewrite the law, however, those days would be over. The first time a developer demonstrated an attractive new product at Comdex or Softeach or some other industry gathering, hundreds of programmers in corporate research laboratories around the world would set to work creating their own versions of the program to compete with the original. A major firm, with a staff of talented programmers, could fairly promptly create a clone of almost any new program and, in so doing, rob the author of much of the value of his creative efforts. The original author might have a head start, but that would provide little comfort once the major firms hit the market with their "new" products. . . . If copying were legal, the creators would lose out. It is as simple as that.

It is no accident that the world's strongest software industry is found in the United States, rather than in some other jurisdiction which provides weaker protection for computer programs. The system is working, and there is no reason to change it.

Plaintiff's Post-Trial Brief at 87-89; *see also* Tarter Decl. ¶¶ 72-73 (Docket No. 289) ("[I]nterface design represents one of the most crucial aspects of software design. In order to promote innovation and the evaluation of better software products, it is important that we protect the investment that the best developers make in interface design."); Clapes, *Software, Copyright, & Competition* at 202-03 (arguing that the user interface, which is often the most important feature of a computer program, requires strong copyright protection to encourage further innovation and advancement).

The opinion testimony offered in this case failed completely to focus on circumstances existing at any time Congress acted to enact or amend any part of the copyright law. For this reason, it has little, if any, value in illuminating issues of law presented in this case. It is relevant, of course, that both the text and history of the copyright law manifest a purpose of encouraging innovation and of doing so through copyright protection. At the time of congressional action, however—even as late as the 1976 Act and the 1980 amendments—it was not possible to know how the computer programming industry would change over the ensuing years, and what exact scope of protection would best advance the policy goals in newly developed and now foreseeably developing circumstances. In light of that uncertainty, shall the court consider expert opinions about what the law ought to be today? What if those opinions appear to be inconsistent with the accommodation, struck by the statute now in effect, among competing and to some extent conflicting public policy justifications for broader or narrower copyright protection?

I conclude that a court would be going well beyond even the precedents most receptive to consideration of legislative history if it took into account expert opinions of persons experienced in the field of computer software development that are manifestly inconsistent with the accommodation that Congress struck when it enacted the Copyright Act of 1976 and the most recent amendments of the copyright law. In the 1976 Act, Congress manifested an intention to encourage innovation in the computer programming field, and to do so

through copyright law that explicitly gives substantial protection to innovative *expression* in intellectual "works," including computer programs. Congress made this decision despite expert testimony that extending copyright protection to computer programs might have "disastrous consequences . . . on standardization in electronic data processing," *Copyright Law Revision: Hearings on S. 597 Before the Subcommittee on Patents, Trademarks, and Copyrights of the Senate Committee on the Judiciary*, 90th Congress, 1st Session 589 (1967) (Testimony of Professor Anthony Oettinger), despite academic criticism of the decision to extend copyright protection to computer programs, Breyer, *Uneasy Case for Copyright*, 84 Harv.L.Rev. at 340-351 (concluding that "[c]omputer programs should not receive copyright protection at the present time"), and despite explicit proposals that copyright protect only "the instructions themselves" while leaving others free to otherwise "replicate [another's] program exactly." *Hearings on S. 597* at 571-73 (Statement of EDUCOM) ("Broad copyright protection for programs is unwise and improper . . . [H]ad programming been constantly carried out under the threat of infringement actions charging plagiarism of existing copyrighted programs, it is doubtful whether the growth of programs and programming techniques of recent years would have been possible."). The 1980 amendments, too, plainly went farther in protecting the interests of the creators of an original software product than defendants' experts now propose (and than some experts at the time proposed, *e.g.*, CONTU, *Final Report* at 56-76 (Hersey, C., dissenting)), even though they did not go as far as proposed by others to whose views Congress then had access.

When a court determines the meaning of the copyright statute in a new context, different from what could have been known to even the most prescient observer at the time Congress acted, the court must respect the congressional mandate—especially with respect to congressional determinations of premise facts—and is constrained to search not for what the court might now independently consider to be the best accommodation of competing interests but instead to deter-

mine an accommodation in this new context that is consistent with the accommodation that Congress enacted for the context to which it spoke. To the extent that expert opinions illumine the comparison between the context to which Congress spoke (the circumstances of software development then existing and predictable future circumstances) and the current context (the changing circumstances of software development now existing and now predictable for the future) the court may and should take them into account. To the extent, however, that expert opinions propose solutions inconsistent with the congressional accommodation, they must be disregarded.

On the basis of the complete record now before me, I conclude that defendants' policy arguments in this case are inconsistent with the accommodation struck by Congress in passing the Copyright Act and its amendments. Moreover, defendants have not demonstrated any significant changes in the premise facts upon which Congress acted, and upon which CONTU rested its recommendations. If anything, user interfaces have become more expressive and more communicative since 1976 and 1980. I thus conclude that I must disregard defendants' experts' prediction of doom for the computer programming industry if copyright is extended to the user interface and other nonliteral elements of computer programs that embody expression, just as Congress disregarded Professor Oettinger's gloomy prediction of "disastrous consequences" and Commissioner Hersey's dissent. Rather, this legal issue must be resolved in such a way as to extend copyright protection, clearly and unequivocally, to those nonliteral elements of computer programs that embody original expression.

#### E. Defendants' Policy Arguments Founded On the OTSOG Principle

One more of defendants' policy arguments deserves attention. Despite statutory mandates supporting the conclusion that elements of the user interface of 1-2-3 are copyrightable, defendants argue that the need to achieve compatibility and standardization compels rejection of that conclusion on pol-

icy grounds. Copyrightability of a user interface, they argue, will frustrate the public interest in allowing programmers to achieve innovation by "borrowing" and improving upon ideas of other programmers, and will undermine attempts to achieve compatibility and standardization among different programs. Especially in the vital area of user interfaces, defendants contend, copyrightability will have adverse consequences on encouraging innovation and on the broader public welfare.

Defendants' general contention—that "Progress of Science and useful Arts" cannot occur unless authors and inventors are privileged to build upon earlier progress and earlier innovation—has long been a virtually unchallenged premise in all branches of the law of intellectual property. An early expression of the point is Newton's declaration: "If I have seen further it is by standing on ye sholders of Giants." Sir Isaac Newton, Letter to Robert Hooke, February 5, 1675/1676, quoted in R. Merton, *On the Shoulders of Giants: A Shan-dean Postscript* 31 (1965).<sup>3</sup> This principle, which Merton has referred to as "OTSOG" (based on the modernized "on the shoulders of giants"), *id.* at 270, is also firmly established in our case law:

In truth, in literature, in science and in art, there are, and can be, few if any, things, which in an abstract sense, are strictly new and original throughout. Every book in literature, science and art, borrows and must necessarily

<sup>3</sup> Newton's now-famous phrase—"on ye shoulders of Giants"—may have been copied, at least in its "total concept and feel," from Bernard of Chartres who, according to Merton, authored the OTSOG aphorism in the early twelfth century when he taught: "we are like dwarfs standing [or sitting, for those dwarfs who prefer safety to far-ranging vision] upon the shoulders of giants, and so able to see more and see farther than the ancients." Merton, *OTSOG* at 178-92. As Merton demonstrates, however, Bernard himself was standing on the shoulders of Priscian, a sixth-century grammarian, who wrote: "The younger [e.g., the more recent] the scholars, the more sharp-sighted." *Id.* at 194-95 (brackets added by Merton). Of course, Priscian's passage presents only the idea; it took Bernard to express that idea in the metaphor of giants and dwarfs.

borrow, and use much which was well known and used before.

*Emerson v. Davies*, 8 F.Cas. 615, 619 (C.C. D.Mass.1845) (No. 4,436) (Story, J.). See also *Ecclesiastes* 1:9-10 ("[T]here is no new thing under the sun. Men may say of something, 'Ah, this is new!'—but it existed long ago before our time."). Defendants, however, attempt to squeeze something from the OTSOG cornerstone that is not there.

Two possible applications of OTSOG in the field of computer programming are relevant to this case. First, innovation in computer programming is advanced as each programmer builds upon the ideas of previous programmers. Second, some of the innovative ideas may be expressed in a particular way that is so effective or efficient that the expression becomes standardized throughout the field even though the idea is capable of being expressed in other ways—that is, even though the particular expression is not an essential detail to every expression of the idea. Although both of these corollaries of OTSOG are important to the future of computer programming, neither was embraced by Congress (as explained below) in such a way as to override the public interest in conferring upon an author a right to a limited monopoly in the author's "work."

The metaphorical "shoulders of giants" on which successors may legally stand are not as broad as defendants contend. The legally relevant shoulders of programming giants are their ideas—and do not extend to all of their expressions. The encouragement of innovation requires no more. It is sufficient that programmers are privileged to borrow and improve upon previous ideas—such as the ideas for an electronic spreadsheet and a two-line moving cursor menu. Adequate room for innovation remains even though successors are barred from copying earlier authors' particular expressions—such as the particular structure, sequence, and organization of a menu command system. *Pearl Systems*, 8 U.S.P. Q.2d at 1525.

Of course, if a previous programmer's idea can be expressed in only one or a limited number of ways—such as the rotated "L" screen display or the use of the slash key to

invoke the menu command system—then the expression, too, may be copied. *Morrissey*, 379 F.2d at 678-79. Where, in contrast, the idea is capable of countless ways of being expressed, only inexpensive cloning, and not innovation, would be advanced by allowing programmers to copy the particular way the ideas have been expressed by others.

The second corollary of the "OTSOG" principle relevant to this case concerns standardization. Defendants have argued that 1-2-3, and specifically, 1-2-3's menu structure and macro command facility, has set a *de facto* industry standard for all electronic spreadsheets. Thus, defendants had no choice, they argue, but to copy these expressive elements from 1-2-3. Had they not copied these elements (including the macro facility), users, who had been trained in 1-2-3 and had written elaborate macros to run on 1-2-3 spreadsheets, would be unwilling to switch to VP-Planner. VP-Planner would be a commercial failure. Neither the factual nor the legal predicate of the argument is supportable.

First, defendants' argument ignores the commercial success of Excel, an innovative spreadsheet program that is not compatible with 1-2-3, either in its menu structure or in its macro command facility. Also, defendants' argument ignores the alternatives to direct copying that were legally available to them.

As already explained, to the extent that VP-Planner was concerned with compatibility for macros written originally for 1-2-3, VP-Planner could have provided for a translation device that could read 1-2-3 macros and convert them, automatically, into macros that could be run on VP-Planner. Microsoft Corporation successfully included such a capability in Excel and Lotus itself has written such a capability for translating macros among different-language versions of 1-2-3. Defendants have not offered persuasive evidence to show that they could not have done the same with VP-Planner. That "[i]t would have been an extremely complicated task" and would have cost defendants more to do so, *Dauphinais Affdvt.* ¶ 163, is not a reason for denial of copyright protection to 1-2-3. Copyright protection always has consequences of this kind.

Moreover, to the extent that VP-Planner is an incremental improvement upon 1-2-3 because of its multidimensional database capabilities, defendants could have (1) sought a license to use 1-2-3's menu structure and command facility; (2) offered to sell their new expression of ideas to Lotus for inclusion in future versions of 1-2-3; or (3) marketed VP-Planner as an "add-in" to 1-2-3.

An "add-in" program is a program designed to be used in conjunction with another program. For example, HAL is a program that is used in conjunction with 1-2-3—a user must have both HAL and 1-2-3—which allows a 1-2-3 user to enter commands in simple sentences such as "copy column B to E" rather than by selecting commands from the menus. Tr. Ex. 176. To encourage the creation of add-in programs (which in turn, make 1-2-3 a more attractive program to potential customers), Lotus has published a product called Lotus Developer Tools that helps developers write add-in programs. Tr. Ex. 138. Thus, even if defendants found the first two alternatives unattractive or unattainable (for example, because of Lotus' nonconsent except on unfavorable terms, *see Stewart*, 110 S.Ct. at 1764 ("a copyright owner has the capacity arbitrarily to refuse to license one who seeks to exploit the work")), they could have marketed their multidimensional database capability as an "add-in" for those users interested in that capability. Transaction and marketing costs of the third option would tend to screen out the new product unless its innovative features made it very attractive to potential users, but this functional effect of copyright law would be entirely consistent with the objects and policies of that law as manifested in the copyright statute and judicial precedents. By instead selling a stand-alone product that completely replaces 1-2-3, defendants have not merely sold and profited from only their incremental additional expression. Rather, they seek permission to profit also from copying Lotus' protected expression.

Defendants' standardization argument is flawed for another reason as well. As explained above, one object of copyright law is to protect expression in order to encourage innovation.

It follows, then, that the more innovative the expression of an idea is, the more important is copyright protection for that expression. By arguing that 1-2-3 was so innovative that it occupied the field and set a *de facto* industry standard, and that, therefore, defendants were free to copy plaintiff's expression, defendants have flipped copyright on its head. Copyright protection would be perverse if it only protected mundane increments while leaving unprotected as part of the public domain those advancements that are more strikingly innovative.

Finally, the entire argument about standardization may be questioned on a more fundamental ground. Defendants have cited no statutory provision or precedent that has ever declared that standardization, when not achieved *de jure*, is necessarily in the public's best interest. The court is aware of no such precedent or legislative mandate. A moment's reflection is enough to disclose that the public interest in extensive standardization is a sharply debatable issue. *See, e.g., Tarter Decl.* ¶¶ 34-45; *cf. the QWERTY typewriter keyboard* (discussed in Farrell, *Standardization and Intellectual Property*, 30 *Jurimetrics* 35, 36-38 (1989)). Decisive against defendants' contention, in any event, is that the particular way they propose that the court resolve this dispute would reduce copyright protection far below the mandate of the copyright act.

## VIII. OTHER DEFENSES

### A. *Lack of Subject Matter Jurisdiction*

Section 411 of Title 17 provides that an infringement action may not be brought unless the author has applied for registration of the copyright. If the application is denied, the infringement action may not be brought unless the author serves a copy of the complaint on the Register of Copyrights. "Copyright registration under § 411(a) is a [jurisdictional] condition precedent to filing an infringement action." *Quincy Cablesystems, Inc. v. Sully's Bar, Inc.*, 650 F.Supp. 838, 850 (D.Mass.1986).

Defendants argue that plaintiff's allegations of copyright infringement in 1-2-3's user interface are, in reality, allegations of copyright infringement in 1-2-3's "screen displays." Defendants further argue that plaintiff's registration certificate for 1-2-3 covers only the literal code of 1-2-3, that plaintiff has failed to register the "screen displays" of 1-2-3 as separate audiovisual works, and that this court consequently lacks jurisdiction over allegations of infringement of the "screen displays" of 1-2-3.

This contention borders on the frivolous. First, I emphatically reject defendants' premise, based on yet another word game, that equates the user interface of 1-2-3 with 1-2-3's "screen displays." In ruling that defendants have infringed plaintiff's copyright, I have not ruled that defendants are liable because they copied the "screen displays" of 1-2-3. Rather, I have concluded that defendants copied protected nonliteral elements of expression in the user interface and the underlying computer program.

Second, I conclude that, in any event, plaintiff's registration certificates are sufficient to extend to the "screen displays" of 1-2-3.

By focusing only on "screen displays," defendants have failed to take due notice of the statutory language as it bears on whether copyrightable elements of computer programs include the "user interface" and other nonliteral aspects.

Depending on the sense in which the phrase "screen display" is used, it may be quite correct to say that

copyright protection of a computer program does not extend to screen displays generated by the program. . . . This distinction results from the fact that the same screen can be created by a variety of separate and independent computer programs. It is somewhat illogical to conclude that a screen can be a "copy" of many different programs.

*Softklone*, 659 F.Supp. at 455-56.

It is not, though, the screen display itself, in this narrow sense, that is a copyrightable "computer program"—that is, "a

set of statements or instructions . . . to bring about a certain result," 17 U.S.C. § 101. The screen display, understood in this narrow sense, is useless for this purpose unless it is integrated with other parts of the computer program so as to become indirectly part of the means by which the user communicates instructions through the total program to bring about the desired result. Thus, the copyrightable element is not, strictly speaking, the screen display, narrowly understood, but the literal and nonliteral elements of not only the display but also the distinctive way of creating it. Indeed, the *Softklone* court recognized this point:

Therefore, it is this court's opinion that a computer program's copyright protection does not extend to the program's screen displays, and that copying of a program's screen displays, without evidence of copying of the program's source code, object code, *sequence, organization, or structure*, does not state a claim of infringement.

659 F.Supp. at 456 (emphasis added).

Accordingly, just as using different dialogue does not sanitize from claims of infringement a play that copies the plot and characters from some other copyrighted work with resulting substantial similarity, the fact that the screen display *could* easily be created in many other ways—either to perform the same or to perform very different functions—does not sanitize it from claims of copyright infringement. The critical question, instead, is whether copyright protection is available against creating and using such a screen display—including the manifestation of structure, sequence, and organization in that display—as a part of the "set of statements or instructions" in a program that is designed "to be used directly or indirectly in a computer in order to bring about a certain result."

In any event, it is appropriate to consider "screen displays" in a broader sense that also includes the structure, sequence, and organization of the underlying program as manifested in the menus presented on the screen displays. I conclude that plaintiff's certificates of copyright registration in the "entire

work" of 1-2-3 are sufficient to extend copyright protection to the "screen displays," in this broader sense, of Lotus 1-2-3. See TX 1-233-501 (Tr. Ex. 1, covering "entire work" of "1-2-3"), TX 1-233-502 (Tr. Ex. 2, covering "entire work" of "1-2-3 Release 1A"), and TX 1-694-602 (Tr. Ex. 3, covering "computer program text" of "1-2-3 Release 2").

The Copyright Office registers different works in different ways. One registration form may be used for nondramatic literary works (Form TX), different forms are required for audiovisual works (Forms PA and VA), and a different form for sound recordings (Form SR). *Compendium of Copyright Office Practices* § 604 (1984) (hereinafter "*Compendium II*"). See also 37 C.F.R. § 202.3(b)(1) (1989). As Congress recognized, however, the seven distinct categories of copyrightable works that are listed as examples in section 102(a) are not mutually exclusive:

The items are . . . overlapping in the sense that a work falling within one class may encompass works coming within some or all of the other categories.

House Report at 53, *reprinted* at 5666. Thus, in registering a work that defies easy categorization, an author is directed to use the application form "in the class most appropriate to the type of authorship that predominates in the work being registered." *Compendium II* § 604 (emphasis added); 37 C.F.R. § 202.3(b)(2). See also *Compendium II* at § 708 ("The appropriate application form is generally determined by the nature of the authorship in which copyright is claimed."); 37 C.F.R. § 202.3(b)(2).

Most computer programs, like 1-2-3, are registered on Form TX as nondramatic literary works. See *Compendium II* at § 702.01 ("computer programs" should be registered as "nondramatic literary works" on Form TX). Some computer programmers, however—notably authors of video games—have instead registered for copyright the screen displays created by the program. *Stern Electronics*, 669 F.2d at 855 (plaintiff "eschewed registration of its program as a literary work and chose instead to register the sights and sounds of

[the video game] as an audiovisual work"); *M. Kramer Manufacturing Co. v. Andrews*, 783 F.2d 421, 442 (4th Cir.1986) ("a copyright in the audiovisual display, which display is created by a computer program, protects not only the audiovisual from copying, but also the underlying computer program to the extent the program embodies the game's expression"). See also, *Atari Games v. Oman*, 888 F.2d at 882 ("Video games, case law confirms, rank as 'audiovisual works' that may qualify for copyright protection."); *Midway Manufacturing Co. v. Artic International, Inc.*, 704 F.2d 1009, 1012 (7th Cir.) (same), *cert. denied*, 464 U.S. 823, 104 S.Ct. 90, 78 L.Ed.2d 98 (1983). This form of registration is particularly appropriate where the chief function, and the bulk of creative expression, of a particular computer program is the creation of a series of pictorial images on the computer screen as with video games.

Other authors of computer programs have applied for dual registrations for computer programs—one "literary" registration to cover the code, and a separate "audiovisual" registration to cover the screen displays. See *Softclone*, 659 F.Supp. at 455-56. This practice has since been rejected by the Copyright Office as duplicative. See Library of Congress, Copyright Office, *Registration of Computer Screen Displays*, Docket No. 87-4, 53 Fed.Reg. 21,817 (1988) ("all copyrightable expression owned by the same claimant and embodied in a computer program, including computer screen displays, is considered a single work and should be registered on a single application form"); Copyright Office, "Computer Programs and Related Screen Displays" (Tr. Ex. 1179) (single registration required for computer programs and their screen displays, including video game computer programs); *Compendium II* at § 607 (single registration covers single unitary work); *id.* at § 609 (copyright owner may "make only one basic registration per work"); 37 C.F.R. § 202.3(b)(6) (1989) (same); *Manufacturers Technologies*, 706 F.Supp. at 993; *Broderbund*, 648 F.Supp. at 1133.

In any event, when Lotus attempted to register separately the screen displays of 1-2-3 as an audiovisual work, the

Copyright Office denied the registration. In a letter to plaintiff, Copyright Examiner Julia Huff explained:

textual screen displays embodied within the computer program that generates them are covered by the registration for the program, without either the need or justification for separate registration for the displays. Because the displays are considered to be an integral part of the program, the authorship in the displays appears to be the same as that contained in the program.

Letter of Jan. 20, 1987, from Huff to plaintiff (Tr. Ex. 1092).

Accordingly, I conclude that Lotus has properly registered 1-2-3 for copyright protection, and that this court does have subject matter jurisdiction over all aspects of this infringement suit.

#### *B. Laches and Equitable Estoppel*

Next, defendants argue that plaintiff is barred from any relief under the equitable doctrines of laches and equitable estoppel.

Defendants have the burden of proving laches. *Costello v. United States*, 365 U.S. 265, 282, 81 S.Ct. 534, 543, 5 L.Ed.2d 551 (1961). Thus, to prevail on this affirmative defense, defendants must prove by a preponderance of the evidence (1) that plaintiff inexcusably and unreasonably delayed in bringing this action; and (2) that the delay has prejudiced defendants. *Gardner v. Panama R. Co.*, 342 U.S. 29, 31, 72 S.Ct. 12, 13, 96 L.Ed. 31 (1951); *Puerto Rican-American Insurance Co. v. Benjamin Shipping Co.*, 829 F.2d 281, 283 (1st Cir.1987). Defendants have failed to sustain either part of that burden.

According to defendants, beginning as early as April 30, 1985, "the trade press trumpeted Paperback's soon-to-be-released spreadsheet product [VP-Planner], making no secret that it would look and work like Lotus 1-2-3." Defendants' Post-Trial Brief at 120. VP-Planner was eventually released on October 30, 1985, but, defendants' charge, Lotus waited until January 12, 1987, 438 days later, to bring suit. *Id.*

Even if Lotus did wait fourteen-and-a-half months after learning of VP-Planner's release before commencing this litigation, I find that this delay was neither unreasonable nor inexcusable. Defendants cite *Worcester Brewing Corp. v. Rueter & Co.*, 157 F. 217, 219 (1st Cir.1907), for the proposition that a fifteen-month delay barred plaintiff's trademark infringement damage claim. That case, though, involved a relatively simple claim that defendant infringed plaintiff's trademark "Sterling Ale." Here, as evidenced by the court's lengthy Opinion, plaintiff's claim of copyright infringement in nonliteral elements of the user interface of 1-2-3 was a far more complicated claim, and one that required much more time to evaluate and prepare.

In circumstances such as these, a delay of fourteen-and-a-half months is not only reasonable, but, in fact, is likely to serve both part and public interests in appropriate use of public and private resources for resolution of disputes. As plaintiff aptly asserts: "Prudent business judgment, Rule 11 and basic common sense required Lotus first to ascertain that the threat to its intellectual property interest was serious and that its legal position was sound before filing suit." Plaintiff's Post-Trial Brief at 83. See Manzi Decl. ¶¶ 8-11 (Docket No. 293); *Roulo v. Russ Berrie & Co., Inc.*, 886 F.2d 931, 942 (7th Cir.1989) (twenty-one-month delay in filing infringement suit reasonable where "delay in filing was due to [plaintiff's] inquiry into the facts to determine the merit of her claim against [defendant]"), *cert. denied*, \_\_\_ U.S. \_\_\_, 110 S.Ct. 1124, 107 L.Ed.2d 1030 (1990).

Defendants nevertheless argue that, during the fourteen-and-a-half-month delay, plaintiff at least had a duty to notify defendants that plaintiff considered VP-Planner objectionable and that plaintiff intended to claim proprietary rights in 1-2-3's user interface. Not only did plaintiff fail to so notify defendants, defendants assert, but, in fact, plaintiff affirmatively led defendants to infer that VP-Planner was not objectionable. Accordingly, defendants contend, plaintiff is estopped from recovering against defendant. Defendants' Post-Trial Brief at 131 n. 66 (citing *Precious Metals Associ-*

*ates, Inc. v. Commodity Futures Trading Commission*, 620 F.2d 900, 908 (1st Cir.1980) (" 'A person is estopped from denying the consequences of his conduct where the conduct has been such as to induce another to change his position in good faith or such that a reasonable man would rely upon the representations made.' ") (in turn quoting *Bergeron v. Mansour*, 152 F.2d 27, 30 (1st Cir.1945))).

In support of this contention, defendants cite a form letter that plaintiff's Supervisor of the Information Center sent to Paperback in January 1986 inquiring as to whether Lotus could use its copy of VP-Planner at any location at plaintiff's business. Tr. Ex. 201. Defendants argue that this letter "suggested that Lotus treated VP-Planner as it treated any other non-objectionable software product." Defendants Post-Trial Brief at 131.

First, defendants have failed to offer any evidence that they did, in fact, draw that inference. Indeed, Steven Cook, Paperback's former president, testified that the form letter was simply returned to plaintiff with the appropriate box checked, and that Paperback did not even retain a copy of the letter. Cook Affdvt. ¶ 24 (Docket No. 299). Second, the court is unable to fathom how any reasonable person could draw that inference from Lotus' form letter. In any event, defendants have cited no authority to support the proposition that Lotus' letter, and Lotus' failure to notify defendants of its internal investigation into possible legal action during the fourteen-and-a-half month hiatus, constitute the kind of actions that support laches or equitable estoppel.

Another reason that defendants have failed to sustain a laches defense is that they have failed to demonstrate, any prejudice.

At the core of defendants' asserted prejudice is the following contention advanced by Paperback's former president:

If Lotus had sued Paperback within three or four months after the release of VP-Planner, then Paperback would not have continued to market, sell, or develop VP-Planner in its Lotus-compatible form, whether or not the litigation had any merit.

Cook Affdvt. ¶ 33.

In light of defendants' belief that their spreadsheet product, to be successful, had to be compatible with 1-2-3, and defendants' strident insistence that 1-2-3's user interface is not copyrightable, this contention is simply not credible. In hindsight, perhaps, the expense of this lawsuit looms immense, but the court does not credit the testimony that the mere threat of a lawsuit, especially one that defendants assert they deemed frivolous, would have caused defendants' to retreat from marketing a 1-2-3 "workalike" spreadsheet program. I find that defendants have not shown that they were prejudiced by any delay.

Accordingly, I find that defendants have failed to sustain either element of a laches defense, or an equitable estoppel defense.

## IX. RESERVED RULINGS ON OBJECTIONS

Although ruling on most evidentiary objections during the trial, the court reserved ruling on some Rule 402 objections until after the court had an opportunity to consider the relevance of certain evidence in the context of the entire case. Most notable is evidence proffered by defendants with respect to VisiCalc, which plaintiff asserted was irrelevant to any issues raised by this case.

Although I agree that the "VisiCalc story," in the decisionmaking process reflected in this Opinion, has turned out to be not relevant to any of the equitable defenses, and not relevant to defendants' contention that Lotus itself copied aspects of VisiCalc with 1-2-3, I conclude that the "VisiCalc story" is relevant for background purposes. See *Granite Music Corp. v. United Artists Corp.*, 532 F.2d 718, 720-21 (9th Cir.1976) (admitting evidence of similar musical phrases appearing in works that predate both plaintiff's and defendant's song proper to show, for background purposes, that certain elements were of such ordinary and common occurrence that the probability of independent coincidental production was great). Also, I conclude that this evidence may appropriately be considered for the purpose of deciding premise

facts upon which the conclusions of law with respect to copyrightability rest. *See generally* Part VII(D), *supra*.

With respect to all other reserved evidentiary rulings, the court concludes that they are moot in view of the findings and rulings stated in this Opinion. Moreover, even if the court were to rule explicitly that all of the evidence was admissible, I find as factfinder that none of this evidence alone, nor all of it taken together, would change any of the findings and conclusions contained in this Opinion. That is, even when taken into account, this evidence has so little probative weight that it does not change any finding. Nevertheless, any party wishing an explicit ruling on any reserved evidentiary objection may so move on or before July 19, 1990, identifying specifically each objection on which a more explicit ruling is requested with references, if available, to specific pages in the transcript of the first phase of this trial.

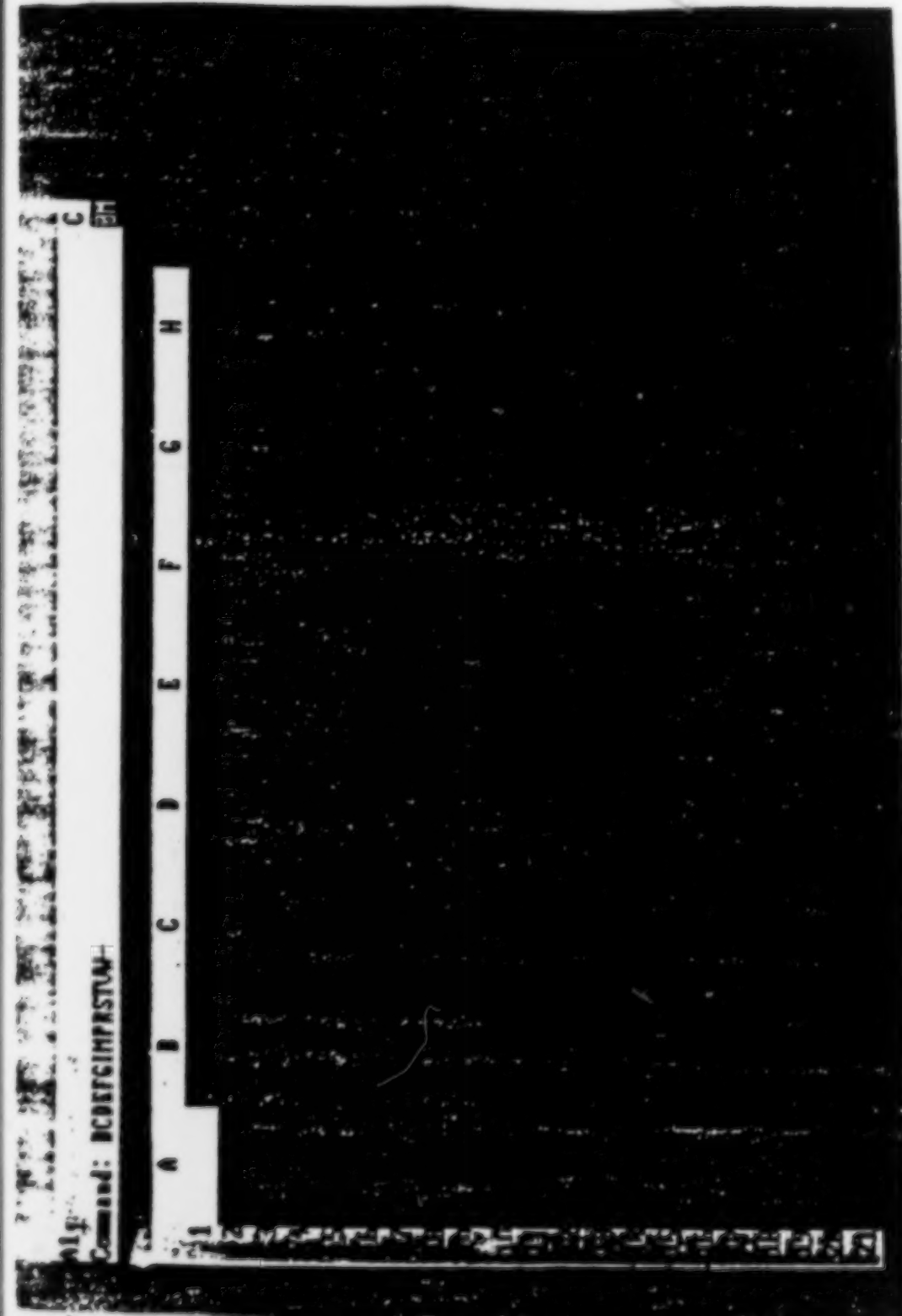
#### ORDER

For the reasons explained in this Opinion and on the findings and conclusions stated, it is hereby ordered:

- (1) Liability for infringement by defendants' is established.
- (2) Any party wishing an explicit ruling on any reserved evidentiary objection may so move on or before July 19, 1990, identifying specifically each objection on which a more explicit ruling is requested with references, if available, to specific pages in the transcript of the first phase of this trial.
- (3) A conference with respect to further proceedings will be held as specified in the Joint Procedural Order of this date.

#### APPENDIX 1

VisiCalc Screen, Main Menu  
Tr. Ex. 1047, Tab A1

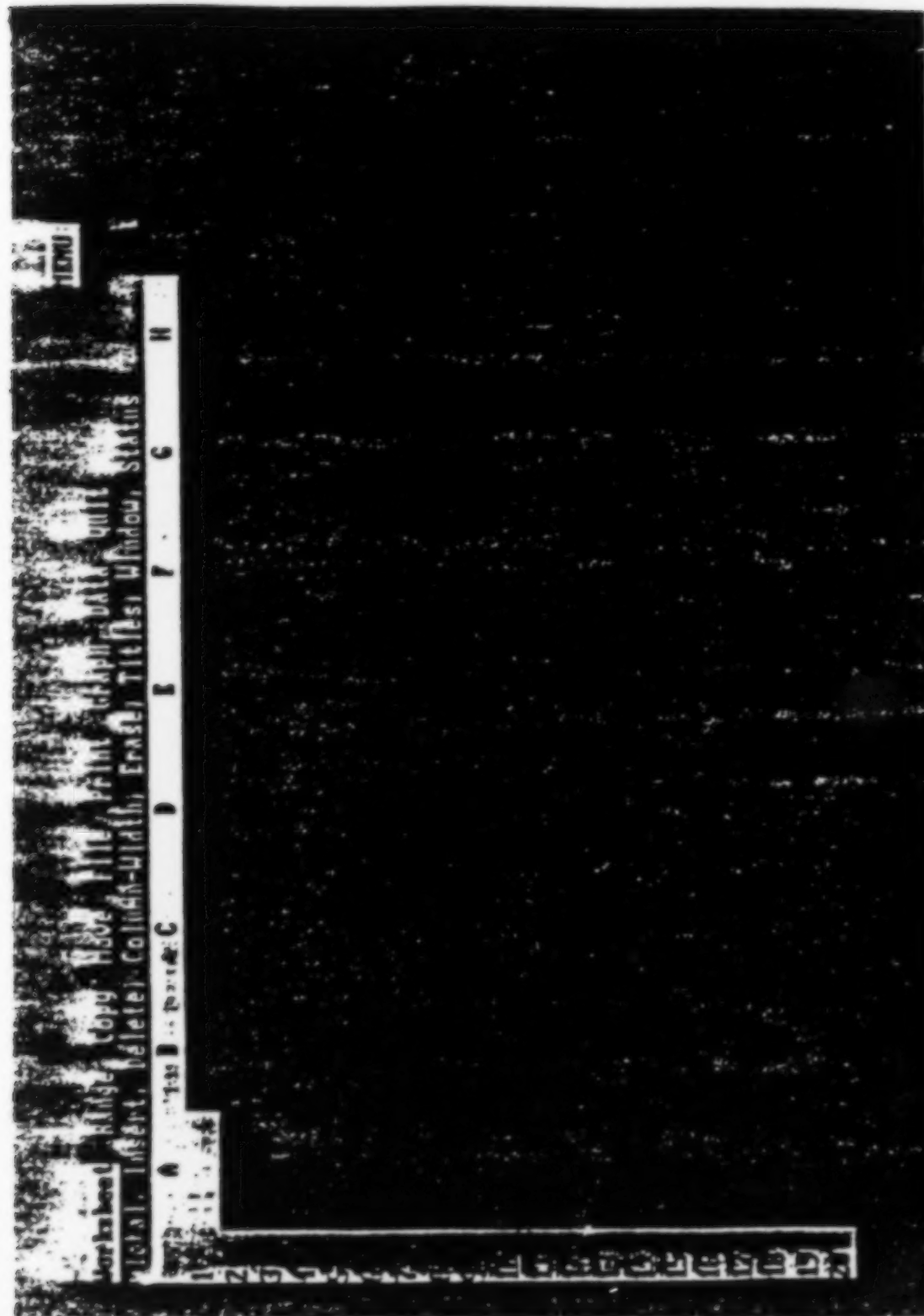


BEST AVAILABLE COPY

268a

# APPENDIX 2

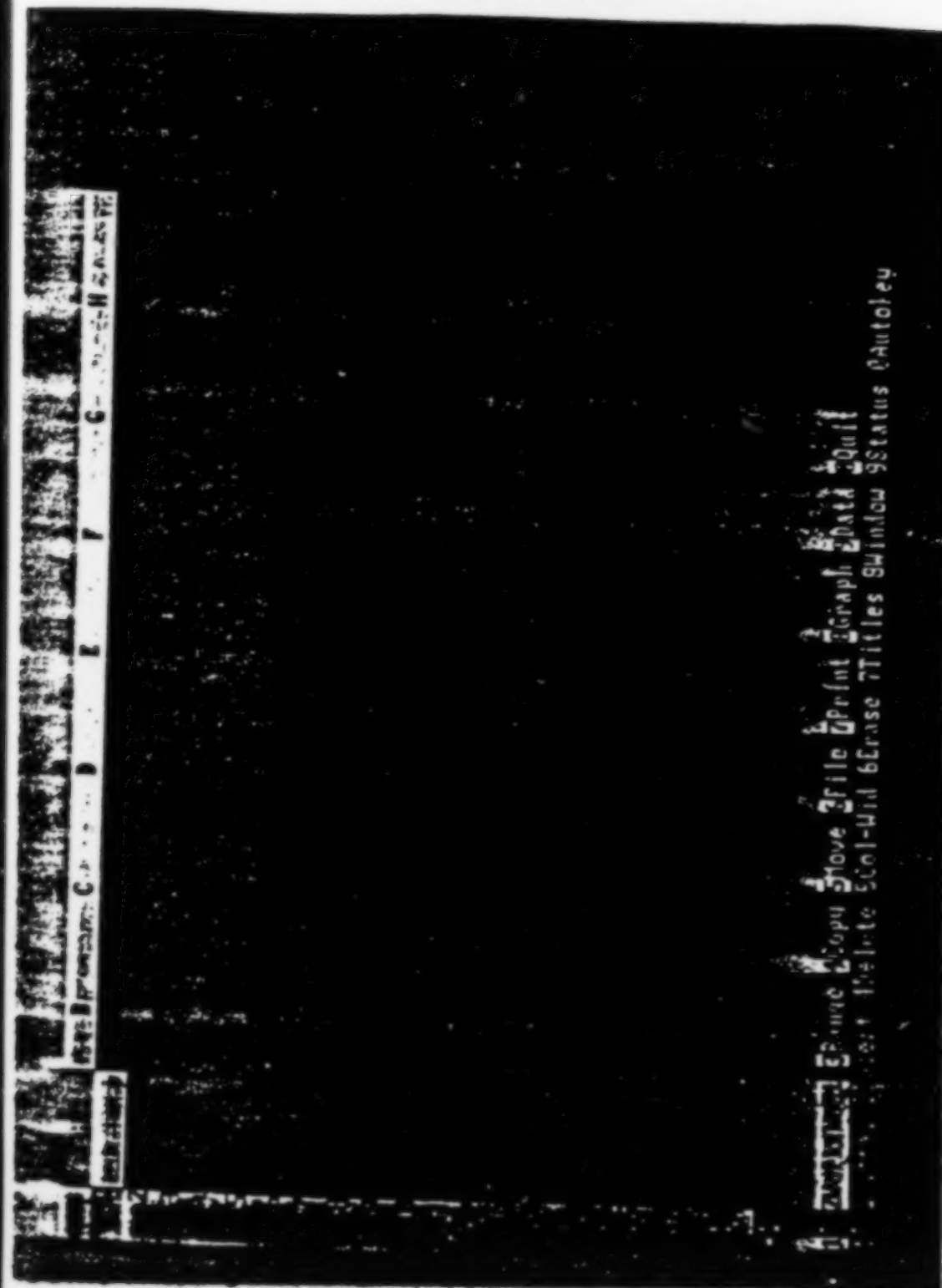
1-2-3 Screen, Main Menu, Worksheet highlighted  
Tr. Ex. 1047, Tab A2



269a

# APPENDIX 3

VP-Planner Screen, Main Menu, Worksheet highlighted  
Tr. Ex. 1047, Tab A4



## COPYRIGHT ACT

## § 101. Definitions

Except as otherwise provided in this title, as used in this title, the following terms and their variant forms mean the following:

An "anonymous work" is a work on the copies or phonorecords of which no natural person is identified as author.

An "architectural work" is the design of a building as embodied in any tangible medium of expression, including a building, architectural plans, or drawings. The work includes the overall form as well as the arrangement and composition of spaces and elements in the design, but does not include individual standard features.

"Audiovisual works" are works that consist of a series of related images which are intrinsically intended to be shown by the use of machines or devices such as projectors, viewers, or electronic equipment, together with accompanying sounds, if any, regardless of the nature of the material objects, such as films or tapes, in which the works are embodied.

The "Berne Convention" is the Convention for the Protection of Literary and Artistic Works, signed at Berne, Switzerland, on September 9, 1886, and all acts, protocols, and revisions thereto.\*

\* Section 2 of the Berne Convention Implementation Act of 1988, Pub. L. 100-568, 102 Stat. 2853 (Oct 31, 1988), provides the following declarations:

The Congress makes the following declarations:

- (1) The Convention for the Protection of Literary and Artistic Works, signed at Berne, Switzerland, on September 9, 1886, and all acts, protocols, and revisions thereto (hereafter in this Act referred to as the "Berne Convention") are not self-executing under the Constitution and laws of the United States.
- (2) The obligations of the United States under the Berne Convention may be performed only pursuant to appropriate domestic law.

(footnote continued)

A work is a "Berne Convention work" if—

(1) in the case of an unpublished work, one or more of the authors is a national of a nation adhering to the Berne Convention, or in the case of a published work, one or more of the authors is a national of a nation adhering to the Berne Convention on the date of first publication;

(2) the work was first published in a nation adhering to the Berne Convention, or was simultaneously first published in a nation adhering to the Berne Convention and in a foreign nation that does not adhere to the Berne Convention;

(3) in the case of an audiovisual work—

(A) if one or more of the authors is a legal entity, that author has its headquarters in a nation adhering to the Berne Convention; or

(B) if one or more of the authors is an individual, that author is domiciled, or has his or her habitual residence in, a nation adhering to the Berne Convention;

(4) in the case of a pictorial, graphic, or sculptural work that is incorporated in a building or other structure, the building or structure is located in a nation adhering to the Berne Convention; or

(5) in the case of an architectural work embodied in a building, such building is erected in a country adhering to the Berne Convention.

For purposes of paragraph (1), an author who is domiciled in or has his or her habitual residence in, a nation adhering to the Berne Convention is considered to be a national of that nation. For purposes of paragraph (2), a work is considered to

(3) The amendments made by this Act, together with the law as it exists on the date of the enactment of this Act, satisfy the obligations of the United States in adhering to the Berne Convention and no further rights or interests shall be recognized or created for that purpose.

have been simultaneously published in two or more nations if its dates of publication are within 30 days of one another.

The "best edition" of a work is the edition, published in the United States at any time before the date of deposit, that the Library of Congress determines to be most suitable for its purposes.

A person's "children" are that person's immediate offspring, whether legitimate or not, and any children legally adopted by that person.

A "collective work" is a work, such as a periodical issue, anthology, or encyclopedia, in which a number of contributions, constituting separate and independent works in themselves, are assembled into a collective whole.

A "compilation" is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship. The term "compilation" includes collective works.

A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

"Copies" are material objects, other than phonorecords, in which a work is fixed by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. The term "copies" includes the material object, other than a phonorecord, in which the work is first fixed.

"Copyright owner", with respect to any one of the exclusive rights comprised in a copyright, refers to the owner of that particular right.

The "country of origin" of a Berne Convention work, for purposes of section 411, is the United States if—

(1) in the case of a published work, the work is first published—

(A) in the United States;

(B) simultaneously in the United States and another nation or nations adhering to the Berne Convention, whose law grants a term of copyright protection that is the same as or longer than the term provided in the United States;

(C) simultaneously in the United States and a foreign nation that does not adhere to the Berne Convention; or

(D) in a foreign nation that does not adhere to the Berne Convention, and all of the authors of the work are nationals, domiciliaries, or habitual residents of, or in the case of an audiovisual work legal entities with headquarters in, the United States;

(2) in the case of an unpublished work, all the authors of the work are nationals, domiciliaries or habitual residents of the United States, or, in the case of an unpublished audiovisual work, all the authors are legal entities with headquarters in the United States; or

(3) in the case of a pictorial, graphic, or sculptural work incorporated in a building or structure, the building or structure is located in the United States.

For the purposes of section 411, the "country of origin" of any other Berne Convention work is not the United States.

A work is "created" when it is fixed in a copy or phonorecord for the first time; where a work is prepared over a period of time, the portion of it that has been fixed at any particular time constitutes the work as of that time, and where the work has been prepared in different versions, each version constitutes a separate work.

A "derivative work" is a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a "derivative work".

A "device", "machine", or "process" is one now known or later developed.

To "display" a work means to show a copy of it, either directly or by means of a film, slide, television image, or any other device or process or, in the case of a motion picture or other audiovisual work, to show individual images nonsequentially.

A work is "fixed" in a tangible medium of expression when its embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration. A work consisting of sounds, images, or both, that are being transmitted, is "fixed" for purposes of this title if a fixation of the work is being made simultaneously with its transmission.

The terms "including" and "such as" are illustrative and not limitative.

A "joint work" is a work prepared by two or more authors with the intention that their contributions be merged into inseparable or interdependent parts of a unitary whole.

"Literary works" are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.

"Motion pictures" are audiovisual works consisting of a series of related images which, when shown in succession, impart an impression of motion, together with accompanying sounds, if any.

To "perform" a work means to recite, render, play, dance, or act it, either directly or by means of any device or process or, in the case of a motion picture or other audiovisual work, to show its images in any sequence or to make the sounds accompanying it audible.

"Phonorecords" are material objects in which sounds, other than those accompanying a motion picture or other audiovisual work, are fixed by any method now known or later

developed, and from which the sounds can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. The term "phonorecords" includes the material object in which the sounds are first fixed.

"Pictorial, graphic, and sculptural works" include two-dimensional and three-dimensional works of fine, graphic, and applied art, photographs, prints and art reproductions, maps, globes, charts, diagrams, models, and technical drawings, including architectural plans. Such works shall include works of artistic craftsmanship insofar as their form but not their mechanical or utilitarian aspects are concerned; the design of a useful article, as defined in this section, shall be considered a pictorial, graphic, or sculptural work only if, and only to the extent that, such design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article.

A "pseudonymous work" is a work on the copies or phonorecords of which the author is identified under a fictitious name.

"Publication" is the distribution of copies or phonorecords of a work to the public by sale or other transfer of ownership, or by rental, lease, or lending. The offering to distribute copies or phonorecords to a group of persons for purposes of further distribution, public performance, or public display, constitutes publication. A public performance or display of a work does not of itself constitute publication.

"Registration," for purposes of sections 205(c)(2), 405, 406, 410(d), 411, 412, and 506(e), means a registration of a claim in the original or the renewed and extended term of copyright."

To perform or display a work "publicly" means—

- (1) to perform or display it at a place open to the public or at any place where a substantial number of persons outside of a normal circle of a family and its social acquaintances is gathered; or

(2) to transmit or otherwise communicate a performance or display of the work to a place specified by clause (1) or to the public, by means of any device or process, whether the members of the public capable of receiving the performance or display receive it in the same place or in separate places and at the same time or at different times.

"Sound recordings" are works that result from the fixation of a sense of musical, spoken, or other sounds, but not including the sounds accompanying a motion picture or other audiovisual work, regardless of the nature of the material objects, such as disks, tapes, or other phonorecords, in which they are embodied.

"State" includes the District of Columbia and the Commonwealth of Puerto Rico, and any territories to which this title is made applicable by an Act of Congress.

A "transfer of copyright ownership" is an assignment, mortgage, exclusive license, or any other conveyance, alienation, or hypothecation of a copyright or of any of the exclusive rights comprised in a copyright, whether or not it is limited in time or place of effect, but not including a nonexclusive license.

A "transmission program" is a body of material that, as an aggregate, has been produced for the sole purpose of transmission to the public in sequence and as a unit.

To "transmit" a performance or display is to communicate it by any device or process whereby images or sounds are received beyond the place from which they are sent.

The "United States", when used in a geographical sense, comprises the several States, the District of Columbia and the Commonwealth of Puerto Rico, and the organized territories under the jurisdiction of the United States Government.

A "useful article" is an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information. An article that is normally a part of a useful article is considered a "useful article".

The author's "widow" or "widower" is the author's surviving spouse under the law of the author's domicile at the time of his or her death, whether or not the spouse has later remarried.

A "work of visual art" is—

(1) a painting, drawing, print, or sculpture, existing in a single copy, in a limited edition of 200 copies or fewer that are signed and consecutively numbered by the author, or, in the case of a sculpture, in multiple cast, carved, or fabricated sculptures of 200 or fewer that are consecutively numbered by the author and bear the signature or other identifying mark of the author, or

(2) a still photographic image produced for exhibition purposes only, existing in a single copy that is signed by the author, or in a limited edition of 200 copies or fewer that are signed and consecutively numbered by the author.

A work of visual art does not include—

(A)(i) any poster, map, globe, chart, technical drawing, diagram, model, applied art, motion picture or other audiovisual work, book, magazine, newspaper, periodical, data base, electronic information service, electronic publication, or similar publication; (ii) any merchandising item or advertising, promotional, descriptive, covering, or packaging material or container; (iii) any portion or part of any item described in clause (i) or (ii);

(B) any work made for hire; or

(C) any work not subject to copyright protection under this title.

A "work of the United States Government" is a work prepared by an officer or employee of the United States Government as part of that person's official duties.

A "work made for hire" is—

(1) a work prepared by an employee within the scope of his or her employment; or

(2) a work specially ordered or commissioned for use as a contribution to a collective work, as a part of a motion picture or other audiovisual work, as a translation, as a supplementary work, as a compilation, as an instructional text, as a test, as answer material for a test, or as an atlas, if the parties expressly agree in a written instrument signed by them that the work shall be considered a work made for hire. For the purpose of the foregoing sentence, a "supplementary work" is a work prepared for publication as a secondary adjunct to a work by another author for the purpose or introducing, concluding, illustrating, explaining, revising, commenting upon, or assisting in the use of the other work, such as forewords, afterwords, pictorial illustrations, maps, charts, tables, editorial notes, musical arrangements, answer material for tests, bibliographies, appendixes, and indexes, and an "instructional text" is a literary, pictorial, or graphic work prepared for publication and with the purpose of use in systematic instructional activities.

(As amended Dec. 12, 1980, Pub. L. 96-517, § 10(a), 94 Stat. 3028; Oct. 31, 1988, Pub. L. 100-568, § 4(a)(1), 102 Stat. 2854-55; Dec. 1, 1990, Pub. L. 101-650, Title VII, § 703, 104 Stat. 5133; June 26, 1992, Pub. L. 102-307, Title I, § 102, 106 Stat. 266.); Oct. 28, 1992, Pub. L. 102-563, § 3, 106 Stat. 4248.)

#### **§ 102. Subject matter of copyright: In general**

(a) Copyright protection subsists, in accordance with this title in original works or authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. Works of authorship include the following categories:

- (1) literary works;
- (2) musical works, including any accompanying words;
- (3) dramatic works, including any accompanying music;
- (4) pantomimes and choreographic works;
- (5) pictorial, graphic, and sculptural works;
- (6) motion pictures and other audiovisual works;
- (7) sound recordings; and
- (8) architectural works.

(b) In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

(As amended Dec. 1, 1990, Pub. L. 101-650, Title VII, § 703, 104 Stat. 5133.)

#### **§ 103. Subject matter of copyright: Compilations and derivative works**

(a) The subject matter of copyright as specified by section 102 includes compilations and derivative works, but protection for a work employing preexisting material in which copyright subsists does not extend to any part of the work in which such material has been used unlawfully.

(b) The copyright in a compilation or derivative work extends only to the material contributed by the author of such work, as distinguished from the preexisting material employed in the work, and does not imply any exclusive right in the preexisting material. The copyright in such work is independent of, and does not affect or enlarge the scope, duration, ownership, or subsistence of, any copyright protection in the preexisting material.

**§ 117. Limitations on exclusive rights: Computer programs\***

Notwithstanding the provisions of section 106, it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:

(1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or

(2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.

(As amended Dec. 12, 1980, P.L. 96-517, § 10(b), 94 Stat. 3028.

\* Section 117 as amended by the Act of Dec. 12, 1980, Pub. L. 96-517 (94 Stat. 3015). Prior to this amendment, Section 117 read as follows:

**§ 117. Scope of exclusive rights: Use in conjunction with computers and similar information systems**

Notwithstanding the provisions of sections 106 through 116 and 118, this title does not afford to the owner of copyright in a work any greater or lesser rights with respect to the use of the work in conjunction with automatic systems capable of storing, processing, retrieving, or transferring information, or in conjunction with any similar device, machine, or process, than those afforded to works under the law, whether title 17 or the common law or statutes of a State, in effect on December 31, 1977, as held applicable and construed by a court in an action brought under this title.

UNITED STATES DISTRICT COURT  
DISTRICT OF MASSACHUSETTS

---

Civil Action No. 87-0076-K

---

LOTUS DEVELOPMENT CORPORATION,

*Plaintiff,*

—vs.—

PAPERBACK SOFTWARE INTERNATIONAL  
and STEPHENSON SOFTWARE, LTD.,

*Defendants.*

---

Civil Action No. 87-0074-K

---

LOTUS DEVELOPMENT CORPORATION,

*Plaintiff,*

—vs.—

MOSAIC SOFTWARE, INC.,

*Defendant.*

## AFFIDAVIT OF MITCHELL D. KAPOR

MITCHELL D. KAPOR, being duly sworn, deposes and says:

1. My name is Mitchell D. Kapor and I reside in Brookline, Massachusetts.
2. I make this affidavit to provide testimony in the Phase I trial of the above-captioned actions.
3. I am the Chairman of the Board and Chief Executive Officer of ON Technology, Inc.

\* \* \*

[\*]<sup>1</sup>

8. In November 1977, I purchased my first personal computer, a Radio Shack TRS-80, and began writing computer programs for use with that computer.

9. In July 1978 I bought an Apple II personal computer and became an independent software consultant, specializing in developing software for the Apple computer.

10. In early 1979 I formed Micro Finance Systems ("MFS"), a computer software development company.

11. By 1979 I had created, designed and written a computer program called Tiny Troll, the first graphics and statistics software package for the Apple II personal computer. Tiny Troll was published by MFS.

12. In 1979 I also enrolled in the Accelerated Masters Program at the Sloan School of Management at MIT, eventually completing three of the requisite four terms of the program.

13. In the fall of 1979, while at the Sloan School, I was approached by the principals of Personal Software, Inc. ("PSI") to inquire whether MFS was interested in creating a software program for PSI to publish, market and sell. I agreed

<sup>1</sup> Bracketed asterisks indicate beginning of page in original document.

to do so. (In addition, during the brief period from approximately April to August of 1980 I was employed by PSI in California).

14. Pursuant to a software distribution agreement between MFS and PSI dated November 3, 1979, I created, designed and wrote two computer software programs, known as VisiPlot and VisiTrend, which were first published, marketed and released for [\*] sale by PSI in the spring of 1981. VisiPlot was a graphics program and VisiTrend was a statistical analysis and forecasting program.

15. Through the Data Interchange Format ("DIF"), a file format and data exchange protocol that I helped design (and which was eventually released to the public by Software Arts, Inc.), users of VisiCalc (a spreadsheet product), VisiPlot and VisiTrend could transfer data between those programs. For example, data stored in files created using VisiCalc could be displayed graphically by transferring the data to, and then using, the VisiPlot program.

16. In the fall of 1981, the software distribution agreement between MFS and PSI was terminated. I sold all rights to the VisiPlot and VisiTrend products to PSI pursuant to an Agreement and Bill of Sale. (A true and correct copy of the Agreement and Bill of Sale is attached as Exhibit A to this affidavit.)

17. In the Agreement and Bill of Sale, I explicitly retained rights to two of my own product concepts, "EBS" and "I-C-L language system and programs". EBS, or Executive Briefing System, a business/professional graphics program, later became Lotus' first product. I-C-L, a concept for an integrated software package that would include spreadsheet, graphics and other capabilities, served as the foundation for a long process of development resulting in the eventual creation of Lotus 1-2-3.

[\*]

18. In January 1982, I founded Professional Software Technology, which in April 1982 was combined with Lotus

Development Company (another company I had created) to form Lotus Development Corporation.

*The Conceptual Origins of Lotus 1-2-3*

19. The original idea for Lotus 1-2-3 came to me in early 1981, while I was developing VisiPlot and VisiTrend. In the course of using DIF to transfer data from VisiCalc to VisiPlot—a cumbersome process—it occurred to me that a product which, unlike these independent, single-purpose programs, would combine spreadsheet and graphing capabilities in a single, integrated program, could have significant market appeal.

20. In the early summer of 1981, I met with Jonathan Sachs ("Sachs"), who was developing various electronic spreadsheet programs. That June, Sachs joined MFS with what I understood to be non-exclusive rights to one of his spreadsheet programs.

21. After Sachs joined MFS, it became our mutual goal and purpose to create an integrated business software product for a personal computer which would be centrally organized around the spreadsheet metaphor, and which would be superior to existing products.

22. The result of our efforts to attain that goal, Lotus 1-2-3, Release 1.0, was first commercially shipped (or "released") on January 26, 1983.

[\*]

23. Throughout the preceding eighteen months, Sachs and I considered hundreds of alternative ideas for virtually every aspect of the product. 1-2-3 ultimately evolved from a careful selection process wherein concepts were often expressed in one form, later disappeared, and then reappeared in another guise—either in 1-2-3 itself or, frequently, in a subsequent Lotus product.

24. The ebb and flow of this continual selection process is reflected in thousands of pages of contemporaneous devel-

opment notes and documents from my files. True and correct copies of some of these notes, handwritten by me during this development period, are attached to this affidavit as Exhibits B through K. These notes are from files that have been maintained, by me or my attorneys since their creation. It was my usual practice during that time to write on the face of my notes the date of their creation.

25. From the outset, I intended to create a product that would have value for, and appeal to, the broadest possible audience. I wanted the product to attract not only experienced computer programmers or users, but also individuals who had no prior computer or spreadsheet experience.

26. To accomplish this, I determined that it was essential to create a product with a user interface dramatically different from those provided in then-existing spreadsheet products—one that would be easy for the inexperienced to learn and [\*] to use, while also providing speed, convenience, and advanced capabilities for the more sophisticated.

27. As my notes of June 18, 1981 (Exhibit B) reflect, early in the development process of 1-2-3 I perceived the user interface as something distinct and separable from the functionality that the spreadsheet program would provide to the user—i.e., the set of operations and functions that the user could perform with the software.

28. Indeed, one of my earliest concepts was to provide a user interface which would permit the user to define his or her own menus of commands, corresponding to the elements of the program's underlying functionality that he or she desired to use. This would, in effect, allow each user to create an individually customized user interface, even if the user had no knowledge of the workings of the program's underlying source code.

29. I had conceived this idea previously in connection with my work on VisiPlot, as my notes of December 29, 1980 (Exhibit C) reflect.

30. This concept, which I came to refer to as a User Interface Generator ("UIG"), is reflected in my notes of March 3 and May 29, 1981 (Exhibits D and E respectively).

#### *Prototype Development*

31. From the summer of 1981 through the end of the year, Sachs and I worked on a prototype for a spreadsheet which

\* \* \*

[\*]

previous concept for allowing users also to define their own menus. (The concept reemerged later in modified fashion.)

40. I determined that a fixed menu structure was more suitable for the novice spreadsheet users to whom, in large part, we sought to appeal, because it would provide a framework in which such users could learn to approach the solution of spreadsheet problems generally. A fixed menu structure would also help to guide the user through the program's capabilities and to select appropriate operations to solve specific application problems.

41. My early thought to include user-defined menus did survive in one feature of the published version of 1-2-3, expressed as the "/XM" command, which allows experienced users to create their own alternative menus.

42. The fixed menu structure I chose to employ incorporated the two-line moving cursor menu (a concept I first introduced in VisiPlot), which permits the invocation of menu commands from predetermined arrays of choices, and which also provides information about those choices to the user, serving in that respect as a kind of on-line reference.

43. The main elements of the two-line moving cursor menu are: (a) words representing a set of available command options displayed in the top line; (b) long prompts displayed in the second, or lower line, containing information concerning the commands displayed in the top line; and (c) the abil-

ity to [\*] review, by moving the cursor along the words in the top line, the information concerning each command option in that menu before selecting any command from that menu.

44. I decided to organize these menus hierarchically, meaning that the selection of one command option from the first level menu could lead in turn to another array of command options on a second level menu (or "submenu"), and so on, branching out in a sequence of descending menu levels. This organization enables the novice user to browse through the menu levels, in order to view the valid sequences of available options (and their corresponding explanations) and to map out a plan for performing a particular task.

45. This hierarchical organization of command menus can be described, or visualized, as a "menu tree". Indeed, the menu structure of Lotus 1-2-3 as released is depicted in the accompanying user's manual in an index entitled, "1-2-3 and PrintGraph Command Trees".

46. The choice of words selected to represent commands and their organization in the menu tree, I believed, would be critical to the success of the user interface, especially for the novice user. I believed that the words should not only inform, but also fit into the overall approach to spreadsheet problems reflected in the menu tree. I also believed that the flow of the menu tree—i.e., its internal logic, moving from level to level—should be easy for the user to understand and reflect as [\*] closely as possible the user's natural way of thinking, so that the user would never have to think like a computer.

47. These beliefs were, in part, derived from my familiarity with previous spreadsheet products, which I felt were deficient in this regard. For example, the alphabet-like menu structures of some existing products, such as VisiCalc, conveyed too little information to the user, required too much memorization and/or cross-referencing to the printed user's manual, and—because they were not, to a meaningful extent, arranged hierarchically—failed to provide the user with much insight into an effective approach to the solution of spread-